

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Alja Kunovar

**Ogrodje za vrednotenje ustreznosti
aktivnosti in orodij pri testiranju
mobilnih aplikacij**

MAGISTRSKO DELO
ŠTUDIJSKI PROGRAM DRUGE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Tomaž Hovelja
SOMENTOR: doc. dr. Damjan Vavpotič

Ljubljana, 2016

Rezultati magistrskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov magistrskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Zahvaljujem se mentorju doc. dr. Tomažu Hovelji in somentorju doc. dr. Damjanu Vavpotiču za pomoč in nasvete pri izdelavi magistrskega dela.

Zahvaljujem se zaposlenim v podjetjih za vloženi čas in sodelovanje v raziskavi.

Hvala Mihi in družini, ki so mi tekom študija in pisanja magistrske naloge vedno stali ob strani ter me spodbujali.

The day we stop exploring is the day we commit ourselves to live in a stagnant world, devoid of curiosity, empty of dreams.

(Neil deGrasse Tyson)

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Pregled ključne literature	5
2.1	Obstoječa ogrodja za vrednotenje procesa razvoja programske opreme	5
2.2	Obstoječa ogrodja za vrednotenje testiranja programske opreme	10
2.3	Značilnosti in dobre prakse pri testiranju mobilnih aplikacij . .	13
3	Razvoj ogrodja	29
3.1	Izhodišča za razvoj ogrodja	29
3.2	Vidiki ocenjevanja	30
3.3	Postopek vrednotenja	35
3.4	Razvrstitev aktivnosti in orodij	39
4	Študija primera	43
4.1	Opis metodologije	43
4.2	Študija primera na izbranih podjetjih	45
4.3	Rezultati študije primera	50
5	Sklepne ugotovitve	85

Seznam uporabljenih kratic

kratica	angleško	slovensko
SDM	Software Development Methodology	Metodologija razvoja programske opreme
DOI	Diffusion of Innovations	Teorija difuzije inovacij
TAM	Technology Acceptance Model	Model sprejetosti tehnologij
TAM2	Technology Acceptance Model 2	Model sprejetosti tehnologij 2
PCI	Perceived Characteristics of Innovating	Zaznane karakteristike inovacij
TPB	Theory of Planned Behaviour	Teorija načrtovanega vedenja
MPCU	Model of Personal Computer Utilization	Model uporabe osebnega računalnika
CMM	Capability Maturity Model	Zmožnostno zrelostni model
CMMI	Capability Maturity Model Integration	Poenoten zmožnostno zrelostni model
TMM	Test Maturity Model	Zrelostni model testiranja
TMMi	Test Maturity Model integration	Poenoten zrelostni model testiranja
...

Povzetek

Naslov: Ogrodje za vrednotenje ustreznosti aktivnosti in orodij pri testiranju mobilnih aplikacij

Mobilne aplikacije so že skoraj desetletje del našega vsakdana. Zaradi vedno večje nasičenosti trga mobilnih aplikacij postaja njihova kakovost ena od glavnih konkurenčnih prednosti. Na kakovost mobilnih aplikacij močno vpliva testiranje, zato je pravilna izbira ustreznih aktivnosti in orodij za testiranje zelo pomembna. V sklopu magistrskega dela je predstavljeno ogrodje za vrednotenje aktivnosti in orodij pri testiranju mobilnih aplikacij, ki vključuje sociološki, tehnični in ekonomski vidik. S tem so bili v proces vrednotenja vključeni vsi ključni deležniki pri procesu testiranja in tako je bila zagotovljena celovitost ogrodja. Ogrodje je zasnovano na podlagi obstoječe relevantne literature. Njegova uspešnost je bila preverjena s pomočjo študije primera v treh izbranih podjetjih, ki se ukvarjajo z razvojem mobilnih aplikacij. Po izvedenih študijah primera je vodstvo podjetij potrdilo, da so z uporabo ogrodja pridobili nove koristne informacije.

Ključne besede: testiranje, mobilne aplikacije, ogrodje za vrednotenje, mobilne tehnologije, izboljšava procesa testiranja, orodja za testiranje.

Abstract

Title: A framework for evaluation of adequate activities and tools for mobile application testing

Mobile applications have been a part of our everyday life for almost a decade. Due to the increasing market saturation of mobile applications, their quality is becoming one of the main competitive advantages. Testing has a major impact on the quality of mobile applications, therefore it is very important to pick adequate activities and tools for mobile application testing. This master thesis presents an evaluation framework for activities and tools used in the testing of mobile applications, which includes social, technical and economic aspects. Considering all three aspects, the evaluation framework involved all key stakeholders in the testing process, which ensured its integrity. The framework was conceived on the basis of existing relevant literature. Its success was verified in three case studies, conducted in companies developing mobile applications. Their managers confirmed that using the framework provided them with new and useful information.

Keywords: testing, mobile applications, framework for evaluation, mobile technologies, testing process improvement, testing tools.

Poglavje 1

Uvod

Dandanes je uporaba metodologij razvoja programske opreme zelo razširjena. Metodologija razvoja programske opreme (ang. *Software Development Methodology – SDM*) je zbirka postopkov, pravil, tehnik, smernic, orodij in dokumentacije, ki razvojni ekipi pomaga pri razvoju programske opreme. Je priporočen način razvoja programske opreme, ki temelji na množici principov, ki podpirajo, upravičujejo in pojasnjujejo priporočila metodologije za določen kontekst [1]. Glavni razlog za razvoj in uporabo metodologij je povečati učinkovitost razvojnega procesa, katerega rezultat je kakovostnejša programska oprema [2].

Del standardnega procesa razvoja programske opreme je tudi testiranje. Za testiranje je znano, da porabi znaten delež sredstev, namenjenih razvoju programske opreme [3], poleg tega pa pomembno vpliva na kakovost razvite programske opreme [4]. Med razvoj programske opreme sodi tudi razvoj mobilnih aplikacij za pametne telefone. Mobilne aplikacije so vse od leta 2007, ko je podjetje Apple predstavilo prvi iPhone, postale del našega vsakdana. To dokazujejo tudi številke, saj je bilo v prvem četrtletju letošnjega leta v dveh vodilnih trgovinah z mobilnimi aplikacijami na razpolago 3,1 milijona mobilnih aplikacij [5] in opravljenih kar 300 milijard prenosov [6]. Med razpoložljivimi aplikacijami je velik delež takih, ki so odstranjene zaradi slabe kakovosti, ki se odraža predvsem v počasnem nalagalnem času, sesutju

aplikacije takoj po zagonu ali slabem uporabniškem vmesniku [7]. Kot je predstavljeno v študiji Muccini et al. [8], se mobilne aplikacije in njihovo testiranje razlikujejo od standardne programske opreme in si zato zaslužijo podrobno preučitev. Pri tem je pomembna pravilna izbira aktivnosti in orodij pri testiranju mobilnih aplikacij. Zato smo v okviru magistrskega dela razvili ogrodje za vrednotenje ustreznih aktivnosti in orodij, ki podjetjem pomaga pri izboljšavi njihovega procesa testiranja pri razvoju mobilnih aplikacij.

Pri izgradnji ogrodja za vrednotenje smo se oprli na že obstoječa ogrodja za vrednotenje procesov razvoja programske opreme. Dosedanji način preučevanja metodologij raziskuje zlasti predpisano uporabo metodologij na nivoju podjetja kot celote, pri čemer pa zanemara dejansko uporabo med razvijalci [9]. Posledica neupoštevanja sociološkega vidika uporabe metodologij je odpor do sprejemanja predpisanih metodologij, kar je eden od ključnih dejavnikov neuspeha projektov razvoja programske opreme [10, 11, 12]. Obstajajo tudi druga ogrodja za vrednotenje, ki poleg sociološkega vidika upoštevajo tudi tehnični in ekonomski vidik. V magistrskem delu smo želeli preučiti vsa relevantna ogrodja za vrednotenje procesov programske opreme in pri tem upoštevati vse omenjene vidike. Poleg omenjenih vidikov je ogrodje pomembno nadgradilo obstoječa ogrodja, saj je prilagojeno specifikam mobilnih aplikacij. Poleg tega je vrednotenje potekalo na ravni posameznih aktivnosti in orodij ter pri tem upoštevalo vse deležnike v procesu testiranja.

Predvideni prispevki in cilji magistrskega dela so torej:

- Pregledati obstoječa ogrodja za vrednotenje procesa razvoja programske opreme in izboljšavo testiranja.
- Pregledati obstoječe dobre prakse pri testiranju mobilnih aplikacij.
- Razviti celovito ogrodje za vrednotenje aktivnosti in orodij pri testiranju mobilnih aplikacij.
- Preveriti uspešnost modela s tremi študijami primera v treh različnih podjetjih, ki se ukvarjajo z razvojem mobilnih aplikacij.

- Analizirati pridobljene rezultate in jih oblikovati v koristne informacije za vodstvo podjetij.

Da bi dosegli zastavljene cilje, smo magistrsko delo oblikovali na sledeči način. V Poglavlju 2 smo pregledali ključno literaturo, ki se navezuje na naše področje. Pri tem smo predstavili obstoječa ogrodja za vrednotenje procesa razvoja programske opreme in izboljšavo testiranj. Preučili smo tudi značilnosti in dobre prakse pri testiranju mobilnih aplikacij ter orodja, ki jih uporabljamo pri testiranju.

V Poglavlju 3 smo predstavili ogrodje za vrednotenje aktivnosti in orodij pri testiranju mobilnih aplikacij. Najprej smo predstavili vse tri vidike, ki smo jih upoštevali pri izgradnji ogrodja, nato pa opisali potrebne korake našega ogrodja za vrednotenje. Poglavlje smo zaključili z opisom razvrstitve posameznih aktivnosti in orodij glede na ocene, pridobljene z ogrođjem.

V Poglavlju 4 smo predstavili metodologijo, ki smo jo uporabili za preizkus našega ogrodja. Predstavili smo študije primera na izbranih podjetjih in rezultate opravljenih študij primera ter predloge podjetjem, kako izboljšati proces testiranja.

V zadnjem Poglavlju 5 smo podali sklepne ugotovitve in spoznanja, do katerih smo prišli pri izdelavi magistrskega dela. Podali smo tudi nekaj usmeritev za nadaljnje delo.

Poglavje 2

Pregled ključne literature

To poglavje zajema relevantno literaturo, ki je pomembna za magistrsko delo. Predstavljena so obstoječa ogrodja za vrednotenje procesov razvoja programske opreme. Ker se naše ogrodje osredotoča na vrednotenje testiranja mobilnih aplikacij, smo pregledali tudi ogrodja, ki se osredotočajo na vrednotenje testiranja programske opreme. Ob tem smo predstavili glavne značilnosti testiranja mobilnih aplikacij in trenutne dobre prakse.

2.1 Obstoječa ogrodja za vrednotenje procesa razvoja programske opreme

Metodologijo razvoja programske opreme opredelimo kot skupek postopkov, pravil, tehnik, smernic, dokumentacije in orodij, ki se uporabijo v procesu razvoja [1]. Uporaba metodologij za razvoj programske opreme je v veliki meri povezana s povečano učinkovitostjo in kakovostjo izdelkov kot tudi z zmanjševanjem potrebnega časa in napora [14]. Številna podjetja si ustvarijo lastno metodologijo. Veliko pa je tudi takšnih, ki obstoječe metodologije prilagodijo ali močno spremenijo, da bolje ustrezajo njihovi organizaciji [15].

Osnova za ogrodja za vrednotenje, ki smo jih predstavili v nadaljevanju, so ogrodja, ki ocenjujejo **stopnjo sprejetosti** metodologij razvoja programske opreme in pokrivajo sociološki vidik sprejetja metodologij. Ena izmed

prvih splošnih teorij, ki razlaga sociološki vidik je Rogersova **teorija difuzije inovacij** (ang. *Diffusion of Innovations* – DOI) [16]. Gre za teorijo, ki poskuša pojasniti, zakaj se določene inovacije med ciljnimi uporabniki razširjajo, določene pa ne. Inovacija, kot jo opredeli Rogers, je ideja, praksa ali predmet, ki ga posameznik zazna kot novost. Na širjenje inovacije vplivajo štirje elementi, in sicer inovacija sama, komunikacijski kanali, čas in družbeni sistem. Za pomoč pri ocenjevanju sprejetosti posameznika je opredeljenih pet dejavnikov inovacij:

- **relativna prednost** (ang. *Relative Advantage*) – predstavlja stopnjo, do katere je inovacija zaznavana kot boljša od predhodnika,
- **skladnost** (ang. *Compatibility*) – predstavlja stopnjo, do katere je inovacija zaznavana skladno z obstoječimi vrednotami, potrebami in preteklimi izkušnjami uporabnikov,
- **kompleksnost** (ang. *Complexity*) – predstavlja stopnjo, do katere je inovacija zaznavana kot težka za uporabo,
- **možnost preizkušanja** (ang. *Triability*) – predstavlja stopnjo, do katere se da inovacijo preizkusiti vsaj v omejenem obsegu,
- **možnost opazovanja** (ang. *Observability*) – predstavlja stopnjo, do katere je rezultat inovacije viden drugim.

Čeprav gre za splošno teorijo sprejemanja inovacij, lahko teorijo uporabimo tudi pri sprejemanju metodologij razvoja programske opreme, saj jo uporabnik zaznava kot novost. Obstajajo tudi druga podobna ogrodja, ki omogočajo ocenjevanje sprejemanja metodologij. V članku Riemenschneider et al. [17] predstavijo in preverijo pet teoretičnih ogrodij, ki se osredotočajo na sprejemanje metodologij s strani razvijalcev. Ogrodja obravnavajo sprejemanje metodologije predvsem s socioloških, psiholoških in kulturnih vidikov, po navadi pa ne upoštevajo poslovnih vidikov, ki so prav tako ključnega pomena za poslovne odločitve [2]. Omenjenih je pet teoretičnih ogrodij:

- **Model sprejetosti tehnologij** (ang. *Technology Acceptance Model* – TAM) - Ogrodje TAM je bilo zasnovano leta 1989 [18, 19] in sodi med najbolj uveljavljena ogrodja za napovedovanje sprejetosti tehnologije s strani posameznika. Ogrodje temelji na teoriji razumne akcije (ang. *Theory of Reasoned Action* – TRA) [20]. Teorija kot glavna dejavnika za odločitve uporabnika o uporabi nove tehnologije navaja:
 - **zaznano uporabnost** (ang. *Perceived Usefulness*) – pove, v kakšnem obsegu posameznik verjame, da bo uporaba informacijskega sistema povečala njegovo delovno uspešnost,
 - **zaznano enostavnost uporabe** (ang. *Perceived Ease of Use*) – pove, v kakšnem obsegu posameznik verjame, da bo uporaba informacijskega sistema enostavna.
- **Model sprejetosti tehnologij 2** (ang. *Technology Acceptance Model 2* – TAM2) – Ogrodje TAM2 [21] nadgradi ogrodje TAM. TAM2 je primeren za sisteme, katerih uporaba je prostovoljna ali obvezna. To so avtorji dosegli z dvema dodatnima dejavnikoma glede na osnovno ogrodje:
 - **subjektivna norma** (ang. *Subjective Norm*) – pove v kakšnem obsegu posameznik predvideva, kakšno ravnanje drugi pričakujejo od njega v določeni situaciji in v kolikšni meri se je njihovim željam pripravljen podrediti,
 - **zaznana prostovoljnost** (ang. *Perceived Voluntariness*) – pove, v kakšnem obsegu uporabniki dojemajo sprejetost odločitve kot neobvezno.
- **Zaznane karakteristike inovacij** (ang. *Perceived Characteristics of Innovating* – PCI) – Ogrodje PCI [22] izhaja iz Rogersove teorije DOI in poleg relativne prednosti, zapletenosti in skladnosti [16] razširja ogrodje še z naslednjimi dejavniki:

- **predstavljaljivost rezultatov** (ang. *Result Demonstrability*) – predstavlja stopnjo, do katere je inovacija zaznavana kot predstavljaljiva,
 - **podoba** (ang. *Image*) – predstavlja stopnjo, do katere inovacija krepi posameznikovo podobo in status v družbi,
 - **vidnost** (ang. *Visibility*) – predstavlja stopnjo, do katere so rezultati inovacije vidni navzven,
 - **prostovoljnost** (ang. *Voluntariness*) – enako kot pri TAM2.
- **Teorija načrtovanega vedenja** (ang. *Theory of Planned Behavior* – TPB) – Teorija načrtovanega vedenja [23] je psihološka teorija, ki uvede tri odločilne karakteristike in s tem povezuje prepričanja z obnašanjem:
 - **zaznavana kontrola vedenja** (ang. *Perceived Behavioral Control*) – se nanaša na dojemanje posameznika o enostavnosti ali težavnosti izvajanja določenega obnašanja,
 - **subjektivna norma** (ang. *Subjective Norm*) – opredeljena enako kot v TAM2,
 - **odnos do obnašanja** (ang. *Attitude Toward the Behavior*) – predstavlja stopnjo, do katere posameznik oceni, ali je obnašanje ugodno ali neugodno.
 - **Model uporabe osebnega računalnika** (ang. *Model of Personal Computer Utilization* – MPCU) – Thompson et al. [24] predstavijo ogrodje, ki vključuje sociološke dejavnike. Ogrodje se nanaša na pozitivne ali negativne čustvene odzive posameznika. Predstavljene so kratkoročne in dolgoročne posledice sprejetja. Pod kratkoročne posledice sodita zahtevnost in ustreznost službe, med dolgoročne pa karijerne posledice, kot so na primer prilagodljivost zamenjave službe in povečanje priložnosti za opravljanje smiselnejšega dela.

Poleg ogrodij, ki ocenjujejo stopnjo sprejetosti metodologij, so pomembna tudi ogrodja, ki ocenjujejo **tehnično ustreznost** metodologij oz. njenih se-

stavnih delov. Tako tehnični kot tudi sociološki vidik sta pomembna, saj organizacije vlagajo veliko sredstev v metodologije razvoja programske opreme. Metodologija se potem lahko izkaže kot tehnično neustrezna za določen projekt ali nesprejeta s strani razvojne ekipe, s čimer postane neuporabna in jo razvijalci lahko zavrnejo. V članku Vavpotič et al. [25] je predstavljeno takšno ogrodje za vrednotenje programske opreme, ki upošteva tako sociološki kot tudi tehnični vidik. Pri upoštevanju sociološkega vidika povečini izhajajo iz omenjenih obstoječih ogrodij (DOI, TAM, TAM2, PCI ...). Posebnost tega ogrodja je tudi to, da pri vrednotenju ne obravnava metodologije kot celote, temveč po njenih sestavnih delih. Sestavne dele predstavljajo elementi metodologije razvoja, kot so aktivnosti, vloge, artefakti, tehnike, predloge, smernice itd. [26]. Ogrodje se je izkazalo kot uspešno, saj je identificiralo tehnično neustrezne in sociološko nesprejete elemente in s tem podalo dragocene informacije podjetjem, kako naj izboljšajo proces razvoja programske opreme. Pomanjkljivost tega ogrodja je, da ne upošteva ekonomskega vidika. Zato je bilo to ogrodje kasneje dodatno razširjeno tudi z ekonomskim vidikom [2]. Do tedaj je veljalo, da je sprejemanje metodologij razvoja programske opreme odločitev, ki jo morajo sprejeti tehnični vodji. Ker pa vpeljevanje metodologije zahteva veliko časa, truda in denarja, je pomembno, da pri sprejemanju odločitev sodeluje tudi vodstvo in uprava podjetja [1]. V ogrodju so sociološki vidik predstavili s karakteristiko pogostost uporabe aktivnosti metodologije razvoja programske opreme v primeru priložnosti za uporabo (FrqUse). Tehnični vidik pa so predstavili s karakteristiko pogostost priložnosti za uporabo aktivnosti. Ena izmed izbranih karakteristik za predstavitev ekonomskega vidika so stroški. Stroški kot izbrana mera za uspešnost podjetja so skladni z večino empiričnih ogrodij, ki preučujejo vpliv informacijskih tehnologij na uspešnost podjetja, vendar zanemarjajo ostale organizacijske, vedenjske in strateške teorije. S tem namenom ogrodje predstavi karakteristiko cilj, ki meri, kako metodologija razvoja programske opreme vpliva na doseganje ciljev organizacije. Kot tretjo karakteristiko uvedejo izdelek, ki ocenjuje, kako metodologija razvoja programske opreme

vpliva na izboljšanje izdelkov in storitev. Z danim ogrođjem so s študijami primera dokazali, da je dodani ekonomski vidik občutno pripomogel k izboljšavi ključnih aktivnosti v procesu razvoja programske opreme.

Z namenom izboljšave procesa razvoja programske opreme se je v 80. letih 20. stoletja začelo z delom na **zmožnostno zrelostnem modelu** (ang. *Capability Maturity Model* – CMM). Model pojasnjuje pomembnost metodologij in meri učinkovitost zmožnosti razvoja programske opreme organizacije. To stori z opredelitvijo petih stopenj zrelosti, ki določajo uspešne temelje za izboljševanje procesov [27]. Kot naslednik CMM je bil razvit **poenoten zmožnostno zrelostni model** (ang. *Capability Maturity Model Integration* – CMMI), ki je integriral in standardiziral posamezne dele CMM-ja ter s tem odpravil njegove pomanjkljivosti [28].

2.2 Obstoječa ogrođja za vrednotenje testiranja programske opreme

V tem poglavju smo pregledali obstoječa ogrođja za vrednotenje testiranja programske opreme. V prejšnjem poglavju opisana modela CMM in CMMI postavljata standarde izboljšav procesa razvoja programske opreme [29]. Zaradi vedno večje zapletenosti programske opreme bodo z zapletenostjo naraščali tudi stroški testiranja. Kot odgovor na pomanjkanje zrelostnih modelov na področju testiranja programske opreme in z namenom izboljšanja procesa testiranja je bil izdelan model, ki je komplementaren modelu CMM. Model se imenuje **zrelostni model testiranja** (ang. *Testing Maturity Model* – TMM) [30]. Model TMM izvira iz modela CMM, zato imata podobno zasnovo, terminologijo in strukturo. Kot njegova nadgradnja je bil predstavljen **poenoten zrelostni model testiranja** (ang. *Test Maturity Model integration* – TMMi) [29]. Ta pozitivno vpliva na kakovost izdelka, učinkovitost testiranja in spodbuja, da testiranje postane v celoti integrirano v razvojni cikel [29]. Podobno kot ostali zrelostni modeli opredeljuje pet zrelostnih stopenj, ki prikazujejo evolucijsko pot k doseganju izboljšanih organizacijskih

procesov:

- **Stopnja 1 Začetna (ang. *Initial*)** – Testiranje je kaotičen in neopredeljen proces in se pogosto obravnava kot del razhroščevanja. Organizacija ne omogoča stabilnega okolja za podporo testiranju, zato je testiranje odvisno od posameznikov. Testiranje se izvaja *ad-hoc* po zaključenem kodiranju. Izdelki so izdani brez ustreznega pregleda nad tveganji in kakovostjo. V veliki meri takšni izdelki ne dosegajo pričakovanj kakovosti in njihov razvoj preseže proračun.
- **Stopnja 2 Upravljana (ang. *Managed*)** – Testiranje postane upravljan proces in je ločen od razhroščevanja, vendar ga deležniki še vedno obravnavajo kot fazo, ki sledi kodiranju. Izdelani so testni načrti, skupaj s pristopi testiranja. Testiranje postane spremljano in nadzorovano. Razdeli se na več stopenj, in sicer na testiranje komponent, integracijsko testiranje, sistemsko testiranje in sprejemno testiranje. Vendar pa se testiranje še zmeraj izvaja relativno pozno v procesu razvoja programske opreme.
- **Stopnja 3 Opredeljena (ang. *Defined*)** – Testiranje je v celoti integrirano v proces razvoja in ni več obravnavano kot faza, ki sledi kodiranju. Testiranje se načrtuje že v zgodnji fazi projekta, kar se dokumentira v testni načrt. Testiranje se uveljavi na ravni organizacije, izvajajo se medsebojni strokovni pregledi (ang. *peer review*). Testiranje se dojema kot poklic in inženirji kakovosti so vključeni v pregledovanje zahtev in specifikacij. Poleg izvajanih testov v stopnji 2 se izvaja še testiranje uporabnosti in zanesljivosti.
- **Stopnja 4 Merljiva (ang. *Measured*)** – Testiranje postane merljiv proces, ki spodbuja nadaljnjo rast. Na ravni organizacije so vpeljane metrike, ki vrednotijo kakovost testnega procesa, ocenjujejo produktivnost in nadzorujejo izboljšave. Testiranje je prisotno skozi celoten proces in nudi podporo pri odločanju poteka projekta. Je podrobno opredeljen proces in kakovost izdelka je razumljena in nadzorovana.

- **Stopnja 5 Optimizirana** (ang. *Optimization*) – Testiranje je v celoti opredeljen in merljiv proces. Organizacija je sposobna nenehno izboljševati procese, ki temeljijo na kvantitativnem razumevanju statistično nadzorovanih procesov. Izboljševanje testnega procesa poteka prek inkrementalnega in inovativnega procesa ter tehnoloških izboljšav. Identificirajo in analizirajo se najpogostejši vzroki za napake ter temu ustrezno se ukrepa, da se napake ne bi ponavljale v prihodnosti.

Model TMMi je namenjen podpori testnim aktivnostim in izboljšanju testnih procesov tako v sistemskem inženirstvu kot tudi v vejah inženirstva razvoja programske opreme.

Prav omenjeni modeli so osnova za ogrodje za vrednotenje testiranja programske opreme, ki ga podaja Farooq [31]. Njegovo izboljšano ogrodje združuje dobre elemente obstoječih ogrodij z upoštevanjem obstoječih dobrih praks in standardov testiranja. Ogrodje je sestavljeno iz šestih komponent:

- Osnovna komponentna
 - **Proces vrednotenja** (ang. *Evaluation Process*) – ukvarja se z vprašanjem, katere informacije potrebujemo za vrednotenje testnega procesa.
- Podporne komponente
 - **Tarča** (ang. *Target*) – ukvarja se z vprašanjem, kateri predmeti testnega procesa so lahko in morajo biti vrednoteni.
 - **Merila** (ang. *Criteria*) – opredeli merila, ki nas zanimajo – uspešnost, učinkovitost, zanesljivost, funkcionalnost, uporabnost.
 - **Tehnike ocenjevanja** (ang. *Assesment Techniques*) – opredeli seznam tehnik ocenjevanja, ki so uporabljene pri vrednotenju testnih aktivnosti, kot so na primer vprašalniki, intervjuji, vpogled v dokumentacijo ipd.

- **Tehnike za združevanje** (ang. *Synthesis Techniques*) – opredeli načine, kako organizirati in združiti podatke ter jih spremeniti v kakovostne informacije glede merjenega procesa.
- **Idealni vzorec** (ang. *Yardstick*) – opredeli idealno sliko testnega procesa, ki ga primerjamo z realno sliko.

2.3 Značilnosti in dobre prakse pri testiranju mobilnih aplikacij

Mobilna aplikacija je vrsta programske opreme, ki je namenjena delovanju na mobilnih napravah, kot so pametni telefoni in tablični računalniki. Razvoj mobilnih aplikacij sodi v razvoj programske opreme. Za testiranje mobilnih aplikacij tako še vedno veljajo določene prakse iz testiranja programske opreme. Vendar pa se mobilne aplikacije in njihovo testiranje v nekaterih vidikih razlikujejo od standardne programske opreme. Eden izmed razlogov za to so raznoliki senzorji in druge funkcionalnosti mobilnih naprav. Sodobnejša mobilna naprava ima namreč poleg kamere, mikrofona in zaslona na dotik še ogromno senzorjev, kot so: senzor ambientne svetlobe, senzor bližine, pospeškometer, žiroskop, magnetni senzor, senzor za tlak, temperatura in vlago [7]. Zaradi omenjenega smo v tem poglavju najprej predstavili osnovne pojme testiranja in nivoje testiranja programske opreme, ki podajo osnovo o procesu testiranja, nato pa predstavili vrste testiranja in vrste testnih orodij. Pri vrstah testiranja smo vključili tako vrste testiranja mobilnih aplikacij kot tudi programske opreme, saj se nekatere prakse testiranja programske opreme uporabljajo tudi pri testiranju mobilnih aplikacij. Pri tem smo dali poudarek na vrste testiranja mobilnih aplikacij. Cilj magistrske naloge ni predstavitev vseh vrst testiranja in testnih metodologij, zato bomo predstavili le tiste, ki se nam zdijo ključne za razumevanje ogroджа za vrednotenje.

2.3.1 Osnovni pojmi testiranja

Poznamo statično in dinamično testiranje programske opreme. **Statično testiranje** je opredeljeno kot proces testiranja brez izvajanja programske kode in se izvaja v procesu preverjanja [32]. **Dinamično testiranje** je definirano kot proces izvajanja programske kode, ki se izvaja v procesu potrjevanja [32]. Zaradi želje po kakovostni programski opremi testne aktivnosti predstavljajo velik delež v procesu njenega razvoja. Farooq [31] v doktorski disertaciji poda novo definicijo procesa testiranja, ki zajema tako statično kot dinamično testiranje, in sicer: *“Proces testiranja (po vsebini) zavzema vse vrste aktivnosti preverjanja in potrjevanja, kakor tudi vse vrste pregledov v vseh fazah testiranja (načrtovanje, projektiranje, izvedba ...) in na vseh nivojih (testiranje enot, integracijsko testiranje ...) aktivnosti testiranja”*. Menimo, da druga definicija bolje povzema proces testiranja in vsebuje vse ključne pojme, ki jih je treba poznati za dobro razumevanje procesa testiranja. Zato bomo predstavili posamezne pojme, ki so navedeni v definiciji:

- **Preverjanje** je proces vrednotenja programske opreme, ki ugotavlja, ali programska oprema v določeni razvojni fazi izpolnjuje pogoje, določene na začetku te faze [33].
- **Potrjevanje** je opredeljeno kot proces vrednotenja programske opreme med ali na koncu razvojnega procesa, ki ugotavlja, ali programska oprema zadostuje določenim zahtevam [33].
- Proces testiranja lahko razdelimo na naslednje **faze testiranja**: planiranje in nadzor, analiza in načrtovanje, implementacija in izvedba, vrednotenje skladnosti programske opreme z izstopnimi merili in poročanje ter zaključevanje testiranja [34].
- Poznamo štiri **nivoje testiranja**, in sicer na testiranje enot, integracijsko testiranje, sistemsko testiranje in sprejemno testiranje. Posamezne nivoje smo podrobneje predstavili v Poglavju 2.3.2.

Testni scenarij
Preveri delovanje nakupov v aplikaciji brez internetne povezave.

Tabela 2.1: Primer testnega scenarija.

V procesu testiranja izvajamo **testne aktivnosti**, kot so na primer regresijsko testiranje, testiranje uporabniškega vmesnika mobilne aplikacije, pisanje testnega načrta ipd. Pri tem uporabljamo določena **testna orodja**, ki se uporabljajo za potrebe testiranja. Takšna orodja so na primer orodja za beleženje hroščev in napak, orodja za izvajanje avtomatskih testov ipd. Več orodij smo predstavili v Poglavju 2.3.4.

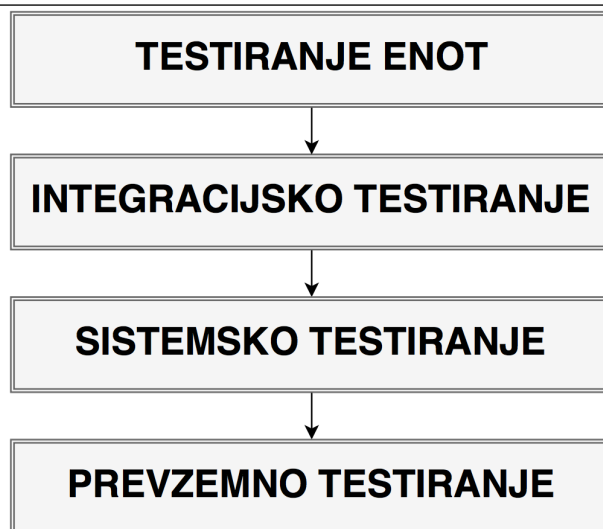
Vsak dober proces mora imeti tudi dober načrt. **Testni načrt** predstavlja dokument, ki opisuje obseg, pristope, vire in urnike za izvajanje določenih testnih aktivnosti. Med drugim navaja testne postavke, funkcionalnosti, ki jih je treba stestirati, testne naloge, navedbo, kdo bo stestiral določeno nalogo, testna okolja in morebitna tveganja [35]. Pri testiranju so ključnega pomena testni primeri in testni scenariji. **Testni scenariji** vsebujejo enovrstične trditve, ki nam povejo, kaj testirati. Trditve so razvrščene glede na funkcionalnosti in so visokonivojske. Iz scenarijev sestavimo več **testnih primerov**, ki predstavljajo nizkonivojske aktivnosti, ki jih je treba izvesti. Podajajo podrobnejše informacije o morebitnih pogojih, kaj testirati, kako testirati in kakšen je pričakovani rezultat [36]. Primera testnega scenarija in testnega primera sta predstavljena v tabelah 2.1 in 2.2.

2.3.2 Nivoji testiranja programske opreme

Kot smo omenili v prejšnjem poglavju, testiranje programske opreme delimo na štiri nivoje [37], ki si sledijo eden za drugim, kot je prikazano na Sliki 2.1. Testne aktivnosti prvih treh nivojev izvajajo člani razvojne skupine, med katero sodijo tudi inženirji kakovosti, medtem ko sprejemno testiranje poteka pri naročniku oziroma ciljnemu uporabniku [38]. Štirje nivoji testiranja so:

Testni primer	Predpogoj	Koraki izvedbe	Pričakovan rezultat	Dejanski rezultat	Status
Nakup v aplikaciji brez internetne povezave	Izključena internetna povezava	1. Odpri aplikacijo.	Aplikacija se pravilno odpre.	Aplikacija se je uspešno odprla.	Opravljeno.
		2. Navigiraj do zaslona, ki prikazuje možne nakupe virtualne valute v aplikaciji.	Prikaže se zaslon z možnimi nakupi.	Zaslon z možnimi nakupi se je uspešno prikazal.	Opravljeno.
		3. Klikni na gumb za nakup tisoč diamantov.	Pokaže se okno "Ni internetne povezave."	Ob kliku se ni zgodilo nič.	Neopravljeno.

Tabela 2.2: Primer testnega primera.



Slika 2.1: Potek nivojev testiranja programske opreme.

- **Testiranje enot** (ang. *Unit Testing*) – Pri testiranju enot se testirajo posamezne enote oz. komponente programske opreme. Enote predstavljajo postopke, metode, funkcije, razrede pa tudi individualne programe. Testiranje enot zagotavlja, da posamezna enota deluje po pričakovanjih. Enote so modularne in neodvisne med seboj, kar nam pogosto omogoča, da iste enote uporabimo večkrat. Testiranje izvajajo razvijalci [39].
- **Integracijsko testiranje** (ang. *Integration Testing*) – Ko so rezultati testiranja enot ustrezni, posamezne enote združimo in jih testiramo skupaj v modulih. Temu pravimo integracijsko testiranje. Z integracijskim testiranjem zagotovimo, da ni nobenih napak med interakcijami v integriranih modulih. Po navadi integracijsko testiranje izvajajo razvijalci [40].
- **Sistemsko testiranje** (ang. *System Testing*) – Sistemsko testiranje je najobsežnejši nivo testiranja in porabi največ sredstev in časa. Na tem nivoju se namreč izvaja testiranje funkcionalnosti (ang. *functional testing*), obremenitveno testiranje (ang. *load testing*), testiranje

zmogljivosti (ang. *performance testing*), testiranje zanesljivosti (ang. *reliability testing*) ipd. Testiranje se izvaja na celotnem in integriranem sistemu z namenom odkrivanja morebitnih napak. Testiranje izvaja testna ekipa [40].

- **Sprejemno testiranje** (ang. *Acceptance Testing*) – Na zadnjem nivoju testiramo sprejetost sistema. Namen testiranja je ovrednotiti, ali je sistem skladen s poslovnimi zahtevami in sprejemljiv za naročnika. Testiranje je najučinkovitejše, če ga izvaja končni uporabnik ali naročnik v okolju, ki najbolje odraža realni svet [40].

2.3.3 Vrste testiranja programske opreme in mobilnih aplikacij

V tem poglavju smo predstavili vrste testiranja mobilnih aplikacij kot tudi programske opreme. Nekatere prakse testiranja programske opreme se namreč uporabljajo tudi pri testiranju mobilnih aplikacij. Razlog za to je v tem, da je mobilna aplikacija ena izmed vrst programske opreme, zato se vrste testiranja medsebojno prepletajo. Pri testiranju mobilnih aplikacij je treba izvajati različne vrste testiranj, da dosežemo določene cilje. Ker je vrst testiranja veliko, smo v nadaljevanju predstavili le tiste, za katere menimo, da so pomembne za razumevanje našega ogrođa.

Regresijsko testiranje

Regresijsko testiranje se uporablja tako pri testiranju mobilnih aplikacij kot tudi pri testiranju programske opreme. Namen regresijskega testiranja je zagotavljanje delovanja aplikacije po spremembah v ključnih delih kode ali ob nadgradnji sistema ali aplikacije. Spremembe kode so lahko izboljšave, refaktorizacija, sprememba konfiguracije, odpravljanje napak ipd. Pogosto se z regresijskim testiranjem iščejo napake, ki so že bile odpravljene, vendar je njihova ponovna pojava zaradi sprememb v kodi zelo verjetna. V idealnem primeru se regresijski testi izvajajo v celoti, vendar smo pogosto

omejeni s časom in viri. Zato je v takih primerih treba narediti analizo vpliva sprememb z namenom identifikacije ključnih področij, ki so lahko prizadeta s spremembami kode in imajo največji vpliv na uporabnike [41].

Funkcionalno testiranje

Namen funkcionalnega testiranja je preverjanje, ali je delovanje določene funkcionalnosti skladno z zahtevami ali tehnično dokumentacijo. Pri testiranju funkcionalnosti ločimo dva pristopa testiranja:

- Testiranje, ki temelji na zahtevah – Testni primeri temeljijo na testnih ciljih in testnih pogojih, ki izhajajo iz zahtev [42].
- Testiranje, ki temelji na poslovnem procesu – Testni primeri so oblikovani na osnovi opisov in poznavanju poslovnih procesov [43].

Testiranje uporabnosti

Testiranje uporabnosti ocenjuje izdelke s pomočjo končnih uporabnikov. S testiranjem uporabnosti se odkrivajo težave, ki jih imajo uporabniki pri uporabi izdelkov in izpostavi možne izboljšave. Testiranje uporabnosti po Nielsonu [44] vključuje naslednjih pet komponent oz. vprašanj:

- Učljivost – Kako preprosto uporabniki opravijo osnovne naloge, ko se prvič srečajo z obliko uporabniškega vmesnika?
- Učinkovitost – Kako hitro lahko izkušeni uporabniki opravljajo naloge?
- Pomnljivost – Ali zna uporabnik aplikacijo po daljši neuporabi učinkovito uporabiti ali se mora aplikacijo ponovno naučiti uporabljati?
- Napake – Koliko napak naredijo uporabniki in kako hude so te napake?
- Zadovoljstvo – Ali je uporabnik zadovoljen z uporabo aplikacije?

Namen testiranja uporabnosti je preverjanje, do kolikšne mere lahko določeni uporabniki aplikacijo uporabljajo za doseganje določenih ciljev z uspešnostjo, učinkovitostjo in zadovoljstvom v določenem kontekstu uporabe [45].

Testiranje zmogljivosti

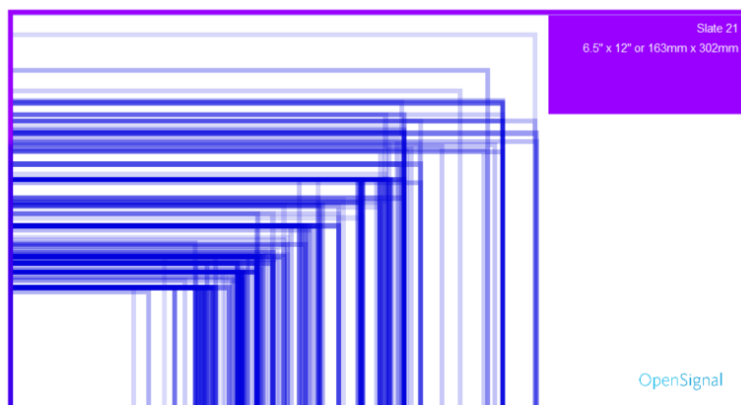
Testiranje zmogljivosti se nanaša na čas izvrševanja določenih funkcij aplikacije, odzivnost aplikacije ter porabo pomnilnika, energije in omrežnih podatkov [46]. Namen testiranja zmogljivosti je identifikacija morebitnega ozkega grla v mobilni aplikaciji. Testiranje je priporočljivo izvajati takoj, ko je to v procesu možno. Za zagotavljanje natančnih rezultatov je treba testiranje izvajati v realnem okolju na dejanskih napravah. Pri testiranju zmogljivosti je treba poleg testiranja same aplikacije vključiti tudi komunikacijo z zalednim sistemom mobilne aplikacije [7].

Testiranje varnosti

Testiranje varnosti je vrsta testiranja, katerega namen je odkriti ranljivost mobilne aplikacije in zagotoviti, da so podatki zaščiteni pred morebitnimi vsiljivci. Testiranje varnosti zagotavlja, da se neavtoriziranim uporabnikom ali procesom onemogoči dostop. Poleg aplikacije same je treba preveriti tudi varnost komunikacije z zalednim sistemom. Najpogostejše varnostne težave pri mobilnih aplikacijah, ki jih moramo preveriti pri testiranju, so predpomnilniki in pomnilniki, kajti v njih ne smemo shranjevati občutljivih podatkov, kot so gesla ali žetoni. Paziti moramo tudi, da so občutljivi podatki in komunikacija šifrirani ter da imajo aplikacije samo tista dovoljenja, ki jih potrebujejo [7].

Testiranje nalaganja mobilne aplikacije

Proces testiranja nalaganja mobilne aplikacije pokriva naložitev, posodobitev in odstranitev mobilne aplikacije. Nalaganje in posodabljanje aplikacije mora potekati brez težav na vseh pomnilnikih in različnih internetnih povezavah. Pri posodabljanju mobilne aplikacije je treba paziti, da se ohrani stanje, kakršno je bilo pred posodobitvijo. Pri odstranjevanju aplikacije je pomembno, da se aplikacija odstrani v celoti in da za sabo ne pušča nerabljenih podatkov.

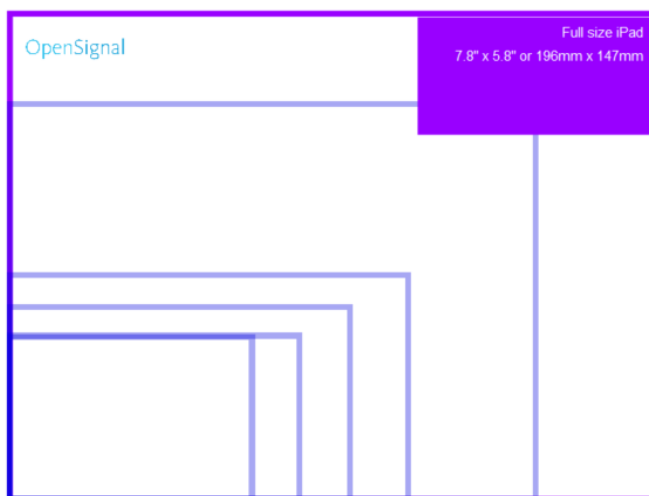


Slika 2.2: Fragmentacija velikosti zaslonov mobilnih naprav Android. Za merilo je prikazan največji zaslon tablice Slate 21, katere diagonala meri kar 54,6 centimetrov [48].

Testiranje uporabniškega vmesnika

Odločili smo se, da bomo testiranje uporabniškega vmesnika obravnavali kot samostojno vrsto testiranja mobilnih aplikacij in ne v sklopu testiranja uporabnosti, kot je zabeleženo v knjigi Daniela Knotta [7]. Testiranje uporabnosti se večinoma ukvarja s tem, ali končni uporabnik zna uporabljati aplikacijo, medtem ko se testiranje uporabniškega vmesnika nanaša na posamezne elemente uporabniškega vmesnika. Pri testiranju uporabniškega vmesnika se ocenjuje, ali so elementi ustrezno postavljeni in ali ustrezajo zahtevam oblikovanja, kot so barve, vrste pisave, velikost pisave ipd. Testiranje uporabniškega vmesnika je pomembno zaradi različnih velikosti zaslonov, saj se morajo elementi uporabniškega vmesnika prilagajati vsem ločljivostim zaslonov. Na Sliki 2.2 so prikazane vse velikosti zaslonov mobilnih naprav Android.

Številčnosti različnih velikosti zaslonov naprav pravimo fragmentacija velikosti zaslona. Manj različnih velikosti zaslonov imajo naprave iOS, vendar se je številka tudi na napravah iOS v zadnjem letu skoraj podvojila. Na Sliki 2.3 so prikazane velikosti zaslonov naprav iOS v letu 2015, vendar na sliki



Slika 2.3: Fragmentacija velikosti zaslonov mobilnih naprav iOS. Za merilo je prikazana tablica iPad, katere diagonala meri 24,6 centimetrov [48].

žal ni velikosti zaslona dveh novejših tablic iOS.

Pri testiranju uporabniškega vmesnika imata pomembno vlogo naslednja pojma:

- **lokalizacija** (ang. *Localization* – l10n) – Nanaša se na prilagoditev mobilne aplikacije jezikovnim, kulturnim in drugim zahtevam na določenem trgu [47].
- **internacionalizacija** (ang. *Internationalization* – i18n) – Načrtovanje in razvijanje mobilne aplikacije mora biti zasnovano tako, da omogoča lokalizacije za ciljne skupine, ki se kulturno in jezikovno razlikujejo [47].

Primer lokalizacije v mobilnih aplikacijah so prevedena besedila, številčni, datumski in časovni formati, denarne enote, tipkovnice, simboli in podobno. Vse to vpliva na uporabniški vmesnik, ki mora biti zaradi lokalizacije prilagodljiv. Ena beseda je namreč lahko v enem jeziku dolga pet znakov, v drugem pa petnajst, zaradi različnih simbolov pa ima lahko tudi različno širino in višino.



Slika 2.4: Elementi uporabniške izkušnje.

Testiranje uporabniške izkušnje

Uporabniška izkušnja je opredeljena kot zaznavanje osebe in njenih odzivov, ki izhajajo iz uporabe ali predvidene uporabe izdelka, sistema ali storitve. Uporabniška izkušnja je sestavljena iz več elementov, kot je prikazano na Sliki 2.4 [49]. Uporabniška izkušnja vključuje vsa uporabnikova čustva, prepričanje, želje, predstave, telesne in duševne odzive, vedenja in dosežke, ki so se pojavljali pred uporabo, med njo in po njej [50]. Zaradi tega je uporabniška izkušnja subjektiven pojem, saj je odvisna od individualnega pogleda posameznika. Kljub subjektivnosti pa je testiranje uporabniške izkušnje zelo pomembno, saj je raziskava pokazala, da je slaba uporabniška izkušnja eden izmed vodilnih vzrokov za neuspeh mobilnih aplikacij [51].

2.3.4 Orodja za testiranje in avtomatizacijo

Za testiranje velja, da je delovno zelo intenzivna naloga. Razlog je v tem, da se testi v veliki meri generirajo, izvajajo in analizirajo ročno. Čas izvajanja takšnih opravil lahko skrajšamo z uporabo ustreznih testnih orodij, kot so na primer generatorji testnih podatkov ali statični analizatorji programske kode. Z uporabo orodij za testiranje povečujemo učinkovitost in uspešnost testiranja. V tem poglavju smo najprej predstavili prednosti avtomatizacije testiranja, nato pa opisali nekaj orodij za testiranje.

Avtomatizacija testiranja

Čas za izvajanje določenih opravil lahko močno zmanjšamo z avtomatizacijo izvajanja testnih aktivnosti. Avtomatizacija pripomore k [38]:

- **Povečanju produktivnosti inženirjev kakovosti** – Pisanje avtomatskih testov omogoča zaposlenim pridobivanje novih znanj in je v primerjavi z ročnim testiranjem manj monotono. Pisanje avtomatskih testov tako zvišuje moralo in posledično povečuje produktivnost.
- **Večji pokritosti regresijskega testiranja** – Količina regresijskih testov se s časom povečuje, zato je pomembno, da se regresijsko testiranje avtomatizira. Avtomatski testi so bolj konsistentni od ročnih testov, kar omogoča lažje reproduciranje testnih rezultatov.
- **Zmanjševanju časa trajanja testiranja** – Zmanjšuje se čas, ki ga potrebujemo za testiranje, kar pripomore k hitrejšemu lansiranju izdelka na trg. Avtomatski testi se lahko izvajajo samodejno izven delovnega časa, s čimer se optimizira uporaba strojne in programske opreme.
- **Zmanjševanju stroškov** – Avtomatizacija znižuje stroške vzdrževanja programske opreme, saj skrajša vire, ki jih potrebujemo za izvajanje regresijskih testov.
- **Večji učinkovitosti testnih primerov** – Avtomatski testi zajemajo tudi ponavljajoče se izvajanje testov, s katerimi lahko odkrijemo puščanje

pomnilnika, težave pri skalabilnosti sistema in lažje izvajamo stresno testiranje.

Pri avtomatizaciji testiranja so prisotna tudi tveganja, ki jih je treba analizirati pred razvojem testov. V primeru slabe analize lahko vpeljava avtomatiziranega testiranja pomeni veliko finančno izgubo. [38] Avtomatizirano testiranje je namreč koristno v primeru večkratnega izvajanja testov, ni pa toliko koristno, kadar se testov ne izvaja pogosto. Zato je treba podrobneje analizirati, kaj se bo avtomatiziralo in kako pogosto se bodo testi izvajali. Treba je tudi poudariti, da avtomatizirano testiranje ne more popolnoma nadomestiti ročnega testiranja. Prav tako ne more nadomestiti človeške ustvarjalnosti, prilagodljivosti in sposobnosti opazovanja [38].

Vrste orodij za testiranje

Za lažje izvajanje testnih aktivnosti uporabljamo orodja. Orodja za testiranje nam lahko pomagajo pri upravljanju celotnega procesa testiranja, na primer kot orodja za upravljanje s testi, zahtevami in hrošči. Pomagajo nam tudi pri izdelavi poročil in nadzorovanju izvajanja testov. Tudi preglednice in urejevalniki dokumentov sodijo pod testna orodja.

Poznamo več vrst različnih orodij za testiranje, ki jih v splošnem razvrščamo v pet vrst testnih orodij, glede na testne aktivnosti, ki jih podpirajo [52]:

- **Orodja za podporo upravljanja testiranja in testov** – Namenjena so nadzoru napredka projekta in kakovosti izdelka.
 - *Orodja za upravljanje testiranja* – Uporabljajo se za upravljanje testnih primerov ročnega in avtomatskega testiranja, testnih okoliš, napak in opravil podjetja [53].
 - *Orodja za upravljanje zahtev testiranja* – So namenjena shranjevanju zahtev in identifikaciji neopredeljenih, manjkajočih ali prihodnjih zahtev testiranja ter pri tem omogočajo sledljivost sprememb [54].

- *Orodja za beleženje incidentov* – Beležijo podatke o napakah, hroščih in drugih anomalijah. Glavne funkcionalnosti orodij so shranjevanje informacij in priponk, prioritizacija incidentov, dodeljevanje nalog, status incidenta ter poročanje o metrikah incidenta [55].
- **Orodja za statično testiranje** – Uporabljajo se za preverjanje skladnosti in programske kode ter algoritmov.
 - *Orodja za statično analizo* – Funkcionalnosti orodja obsegajo identifikacijo kršitev dobrih praks pisanja programske kode, podajajo metrike programske kode in preverjajo strukturo kode [57].
 - *Orodja za modeliranje* – Generirajo vhodne podatke ali celotne testne primere iz shranjenih podatkov o posameznem modelu. Identificirajo neskladnosti in defekte znotraj modelov ter predvidevajo sistemske odzive in obnašanja pod določenimi pogoji [58].
- **Orodja za opredelitev testov**
 - *Orodja za načrtovanje testov* – Uporabljajo se za načrtovanje testnih primerov ali vsaj vhodnih podatkov, ki so del testnega primera. [59]
 - *Orodja za pripravo testnih podatkov* – Omogočajo izbiro podatkov iz obstoječe podatkovne baze in njihovo pripravo za potrebe testiranja ter generiranje novih podatkov. Orodja so koristna pri testiranju zmogljivosti in zanesljivosti, ko je potrebna velika količina realnih podatkov [60].
- **Orodja za izvajanje in beleženje testiranj**
 - *Orodja za izvajanje testov* – Omogočajo izvajanje testov na podlagi testne skripte. Prednost testnih skript je ponovljivost testov z različnimi vhodnimi podatki. Pogosto se izvajajo v sklopu regresijskega testiranja [61].

- *Orodja za testiranje enot* – Orodja omogočajo testiranje, identificiranje in lociranje defektov v kodi ter omogočajo podporo za objektno orientirano programsko opremo. Orodja za testiranje enot shranjujejo testne primere in nadzorujejo, ali je bil test opravljen uspešno ali ne [62].
- *Primerjalniki testov* – Orodja omogočajo avtomatizacijo primerjave med dejanskim in pričakovanim rezultatom testiranja programske opreme. Ločimo dinamično primerjavo, pri kateri se primerjava izvaja med testiranjem, in primerjavo po izvedbi, ki se izvede šele po opravljenem testiranju [63].
- *Orodja za merjenje pokritosti* – Orodja pomagajo pri ocenjevanju temeljitosti testiranja programske opreme. Orodje identificira elemente, ki so bili sproženi, jih prešteje, nato izračuna odstotek pokritosti in na koncu poda seznam nepokritih elementov. [64]
- *Orodja za testiranje varnosti* – Orodja poskušajo vdreti v sistem. Orodja pomagajo pri identifikaciji virusov, napadov za zavrnitev storitve (ang. *Denial of Service Attacks*), slabosti gesel ter pri preverjanju integritete datotek.[65].

• **Orodja za testiranje zmogljivosti in nadzorovanje**

- *Orodja za dinamično analizo* – Orodja omogočajo analizo kode med delovanjem programske opreme. Funkcionalnosti orodja obsegajo zaznavanje puščanja pomnilnika, aritmetičnih napak in podobno. Orodja se uporabljajo na nivoju testiranja enot in integracijskega testiranja [66].
- *Orodja za testiranje zmogljivosti* – Orodja omogočajo podporo obremenitvenemu testiranju (ang. *load testing*), količinskemu testiranju (ang. *volume testing*) in stresnemu testiranju (ang. *stress testing*). Z obremenitvenim testiranjem preverjamo obnašanje sistema pri veliki količini podatkov. S količinskim testiranjem preverjamo obnašanje sistema pri velikem številu uporabnikov. S

stresnim testiranjem pa testiramo meje sistema pri veliki količini podatkov in uporabnikov ter z omejenimi viri, kot je na primer pomnilnik [67].

- *Orodja za nadzor* – Omogočajo nenehno beleženje stanja sistema v uporabi, z namenom opozarjanja v primeru napake ali odpovedi sistema [68].

Poglavje 3

Razvoj ogrodja

V prejšnjem poglavju smo se seznanili z obstoječimi ogrodji vrednotenja metodologij programske opreme in njihovimi pomanjkljivostmi. V tem poglavju smo na osnovi pregleda literature zgradili celovito ogrodje za vrednotenje aktivnosti in orodij pri testiranju mobilnih aplikacij ter naslovili omenjene pomanjkljivosti. Opisali smo tri vidike, ki smo jih uporabili pri izgradnji našega ogrodja ter opisali postopek vrednotenja in razvrstitve aktivnosti in orodij.

3.1 Izhodišča za razvoj ogrodja

Kot smo omenili v prejšnjem poglavju, obstoječa ogrodja po navadi obravnavajo metodologijo kot celoto. Težava je v tem, da s takšnim načinom ne prepoznamo neustreznih delov znotraj metodologije. Ni namreč nujno, da je celotna metodologija neustrezna in potrebna menjave. V našem ogrodju smo metodologijo obravnavali po delih, in sicer po aktivnostih in orodjih, katerih izvajanje in uporabo zahteva. Na ta način smo lahko identificirali posamezne neustrezne dele metodologije in zanje predlagali izboljšave, ustrezne dele pa pustili nedotaknjene. V proces vrednotenja smo vključili vse ključne deležnike pri procesu testiranja, in sicer vodstvo in upravo, tehnični vodji ter razvojno ekipo, v katero poleg razvijalcev sodijo tudi skrbniki kakovosti.

Z vključitvijo vseh ključnih deležnikov smo odpravili pomanjkljivosti nekaterih omenjenih modelov, ki metodologijo vrednotijo samo z enega vidika. Raziskave so namreč pokazale, da je lahko metodologija tehnično ustrezna, vendar ni sprejeta s strani uporabnikov. Zaradi tega naše ogrodje upošteva tako tehnični kot tudi sociološki vidik. Ker so raziskave pokazale, da je pomembno, da pri sprejemanju metodologije sodeluje tudi vodstvo in uprava, smo vključili tudi ekonomski vidik. Z omenjenim bomo zagotovili celovitost našega modela.

Pri vrednotenju procesa smo poleg aktivnosti in orodij, ki se nanašajo izključno na testiranje mobilnih aplikacij, zajeli tudi aktivnosti in orodja ostalih faz procesa, kot jih opredeli RUP (ang. *Rational Unified Process*) [4]. Z vključitvijo aktivnosti in orodij ostalih delov razvojnega procesa smo lahko ocenili posamezne aktivnosti in orodja testiranja relativno za celoten razvojni proces. Kljub temu smo pri zajemu aktivnosti in orodij dali večji poudarek aktivnostim in orodjem, ki se nanašajo na testiranje mobilnih aplikacij.

3.2 Vidiki ocenjevanja

V tem poglavju smo predstavili posamezne vidike ocenjevanja. Kot smo že omenili, so to sociološki, tehnični in ekonomski vidik. Obravnavali smo več vidikov, saj s tem povečamo stopnjo veljavnosti, točnosti in zanesljivosti modela [69]. Posamezne vidike smo nato razbili na karakteristike, ki smo jih opisali v nadaljevanju.

3.2.1 Sociološki vidik

Sociološki vidik upošteva družbene in sociološke lastnosti zaposlenih ter s tem ocenjuje sprejetost posameznih aktivnosti in orodij s strani njenih uporabnikov. V našem ogrodju bo sociološki vidik ocenjevala razvojna ekipa, med katero sodijo razvijalci mobilnih aplikacij in skrbniki kakovosti. Pri ocenjevanju sprejetosti metodologije smo se oprli na obstoječa ogrodja sprejetosti metodologij, ki smo jih predstavili v poglavju Pregled literature in ključnih

področij. Pri opisu karakteristik sociološkega vidika smo aktivnosti in orodja zajeli s skupno besedo element.

Karakteristike sociološkega vidika:

- **Pogostost uporabe, če se pojavi priložnost** – Pove, kako pogosto uporabniki uporabljajo element, če se pojavi priložnost za njegovo uporabo. Karakteristika izhaja iz ogrodja [25] in je nadgradnja karakteristike *pogostost uporabe*. V različnih stopnjah razvojnega procesa so namreč priložnosti za uporabo elementov različne. S pogostostjo uporabe, če se pojavi priložnost, zagotovimo enakost vseh elementov. S to karakteristiko ugotovimo, ali bo zaposleni uporabil element, če se mu pri delu ponudi priložnost za uporabo. Tako identificiramo elemente, ki jih zaposleni manj uporabljajo in so morda nesprejeti.
- **Relativna prednost** – Pove, v kakšnem obsegu se uporabniku zdi, da mu uporaba elementa olajša, pohitri ali izboljša delo. Izhaja iz obravnavane Rogersove teorije DOI. Ker velja za eno najpomembnejših karakteristik, ki pozitivno vplivajo na sprejetost metodologije [16, 17], smo jo uporabili tudi v našem ogrodju. Karakteristika primerja, ali uporaba elementa zaposlenemu na kakršen koli način olajša delo, kot če zaposleni elementa ne bi uporabil.
- **Dostopnost do znanja** – Pove, v kakšnem obsegu je dosegljivo znanje, ki ga potrebujejo uporabniki za uporabo določenega elementa. Zaposleni lahko pridobivajo ustrezno znanje iz različnih virov, kot so na primer delavnice, knjige, spletni viri ipd. S karakteristiko bomo identificirali elemente, ki so nesprejeti, saj zaposleni nimajo na voljo dovolj virov, s katerimi bi lahko dostopali do znanja. Karakteristika je povzeta po članku Vavpotič et al. [25].

3.2.2 Tehnični vidik

Tehnični vidik ocenjuje tehnično ustreznost posameznih aktivnosti in orodij. Tehnični vidik smo podrobneje predstavili že v poglavju Pregled literature in

ključnih področij. Ocenjujejo ga tehnični vodji. Ti imajo namreč ustrezno tehnično znanje za ocenjevanje posameznega elementa, prav tako pa imajo pregled nad celotnim procesom. Pri ocenjevanju tehnične ustreznosti smo uporabili nekaj karakteristik tehničnega vidika iz ogrodja [25]. Tudi tukaj smo aktivnost in orodje zajeli s skupno besedo element.

Karakteristike tehničnega vidika:

- **Skladnost s standardi informacijskih tehnologij** – Predstavlja stopnjo, do katere je določen element skladen z informacijskimi tehnologijami in notranjimi standardi organizacije oz. podjetja. Karakteristiko smo prilagodili potrebam našega ogrodja in tako glede na ogrodje [25] združili dve karakteristiki. S karakteristiko bomo ocenjevali tako skladnost elementa z notranjimi standardi, kar nam bo podalo oceno o notranji celovitosti metodologije, kot tudi splošne standarde informacijskih tehnologij, kot so na primer standardi kodiranja in arhitekture.
- **Vpliv na organizacijo** – Pove, kakšen vpliv ima element na strateške cilje in ugled organizacije oz. podjetja. S karakteristiko se ugotovi, ali element pozitivno vpliva na kakovost procesa in ali je s tem tehnično ustrezen.
- **Skladnost s tehničnimi potrebami projekta** – Pove, do katere stopnje je element skladen s tehničnimi potrebami oz. zahtevami projekta. S to karakteristiko se ugotovi, ali je možno posamezni element prilagoditi velikosti, zahtevnosti, vrsti projekta (nov razvoj, posodobitev) in prioritetam projekta.

3.2.3 Ekonomski vidik

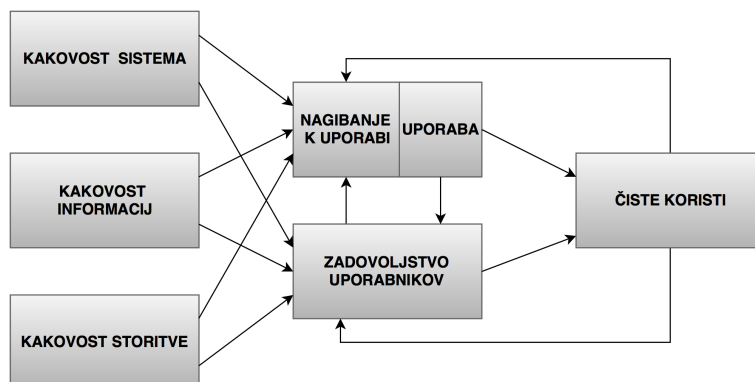
Za celovitost ogrodja ni dovolj, da ocenjujemo le sociološki in tehnični vidik. Pomemben je tudi pogled vodstva in uprave podjetja, saj njih bolj kot zadovoljstvo uporabnikov in tehnična ustreznost zanimajo ekonomske koristi. Zato so nekatera ogrodja [2] dodala tudi ekonomski vidik, ki podaja oceno ekonomske učinkovitosti.



Slika 3.1: Model uspešnosti projektov kvadratna pot [70].

Ekonomska uspešnost projektov programske opreme se je v preteklosti najpogosteje ocenjevala s t. i. železnim trikotnikom (ang. *iron triangle*). Železni trikotnik predstavlja klasična merila uspešnosti, in sicer stroške, čas in kakovost. Kljub svoji razširjenosti, pa je bil deležen številnih kritik [70]. Kritiki izpostavljajo, da se čas in stroški izračunavajo v fazi načrtovanja projekta, ko je skoraj nemogoče določiti realistične omejitve in cilje, kakovost pa opiše kot fenomen, ki se med življenjskim ciklom projekta nenehno spreminja. Na podlagi kritik Atkinson [70] predlaga izboljššan model, pri katerem podjetja poleg železnega trikotnika uporabijo še druge karakteristike, s katerimi lahko merimo uspeh. Izboljšani model se imenuje kvadratna pot (ang. *square route*) in je predstavljen na Sliki 3.1. Del modela, in sicer koristi deležnikov, smo uporabili kot osnovo za karakteristike našega ogrodja.

Poleg omenjenega ogrodja smo uporabili **posodobljeni model uspešnosti informacijskih sistemov DeLone in McLean** [71], ki ocenjuje informacijske sisteme na podlagi kakovosti informacij, sistemov in storitev. Kakovost informacij, sistemov in storitev vpliva na uporabo sistema oziroma na nagibanje k uporabi ter v primeru pozitivne izkušnje tudi na zadovoljstvo



Slika 3.2: Posodobljen model uspešnosti informacijskih sistemov DeLone in McLean [71].

uporabnikov. Kot rezultat uporabe in zadovoljstva uporabnikov se pojavijo čiste koristi (ang. *net benefits*). Čiste koristi lahko opredelimo kot razliko med koristmi, ki jih informacijski sistemi prinašajo podjetjem, ter stroški, ki jih imajo podjetja z razvojem in uporabo teh informacijskih sistemov [72].

Uporaba informacijskega sistema ali storitve je smiselna takrat, kadar so čiste koristi pozitivne. Pozitivne čiste koristi spodbujajo uporabnika k ponovni uporabi sistema ali storitve in povečujejo zadovoljstvo uporabnikov. Poznamo tudi negativne čiste koristi. V primeru negativnih čistih koristi so uporabniki vedno manj zadovoljni in bodo sistem ali storitev uporabljali manj pogosto ali pa bodo sistem ali storitev v celoti prenehali uporabljati. Model in podrobnejše povezave med njegovimi dimenzijami so predstavljene na Sliki 3.2. Iz modela smo za naše potrebe uporabili karakteristiko čiste koristi, kot jo vidita vodstvo in uprava podjetja.

Karakteristike iz obeh omenjenih modelov smo izbrali tako, da skupaj predstavljajo celoto, saj zastopajo interes vseh ključnih deležnikov, in sicer interes vodstva in uprave, interes razvojne ekipe ter interes kupcev oz. naročnikov.

Karakteristike ekonomskega vidika:

- **Čiste koristi** – Pove, v kolikšni meri izvajanje elementa poveča čiste

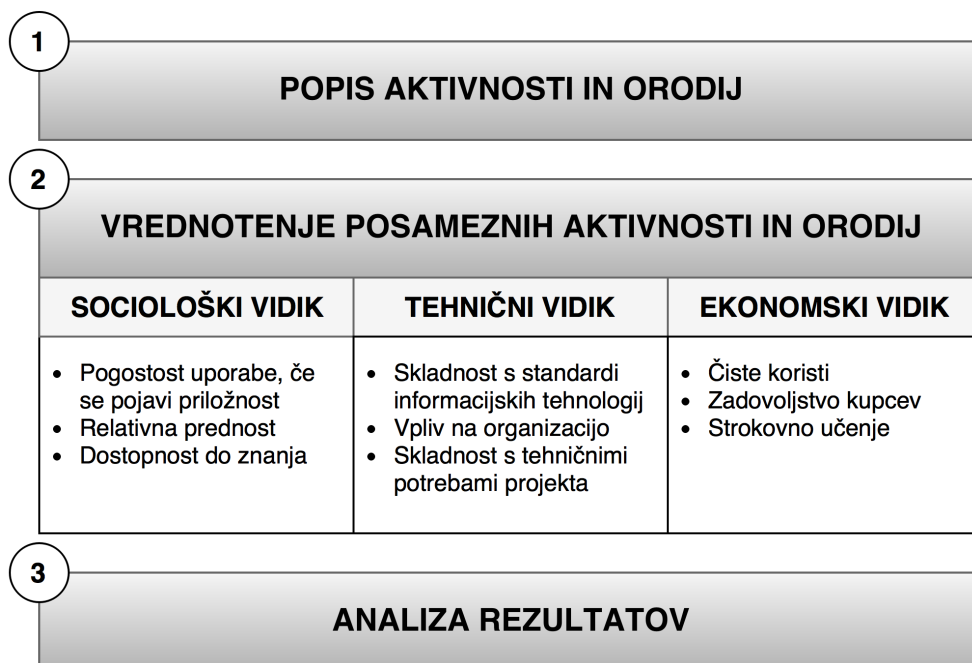
koristi. Kot smo omenili zgoraj, karakteristika izhaja iz posodobljenega modela uspešnosti informacijskih sistemov DeLone and McLean. Pri opredelitvi čistih koristi smo upoštevali interes vodstva in uprave podjetja, ki je čiste koristi opredelilo kot: izboljšano odločanje, zmanjšanje stroškov, razširjenost na trgu, prihranek časa, povečanje prodaje, izboljšana produktivnost.

- **Zadovoljstvo kupcev** – Pove, v kakšni meri izvajanje elementa izpolnjuje kupčeve oz. naročnikove zahteve. Karakteristiko najdemo v obeh obravnavanih modelih. Karakteristika zastopa interes naročnikov in ima posledice za prihodnjo uspešnost podjetja.
- **Strokovno učenje** – Pove, v kakšnem obsegu izvajanje elementa zaposlenim omogoča pridobivanje novih znanj. Karakteristika zastopa interes razvojne skupine. Nova znanja dolgoročno pripomorejo k boljšemu poslovanju podjetja.

3.3 Postopek vrednotenja

V tem podpoglavju smo predstavili tri korake našega ogrodja za vrednotenje, ki je predstavljeno na Sliki 3.3.

V prvem koraku je treba dobiti vpogled v delovanje podjetja, njegove procese in zaposlene, ki te procese uporabljajo. Najprej je treba identificirati vloge v podjetju ter jih razdeliti na ključne deležnike, kot so vodstvo, tehnični vodja in razvojna ekipa. Sledi podrobnejša analiza procesa razvoja mobilnih aplikacij, v katerega sodi tudi testiranje. Analiza se opravi z izvajanjem intervjujev s ključnimi deležniki ter pregledovanjem obstoječe dokumentacije, iz česar pridobimo informacije o aktivnostih, ki se izvajajo, in o uporabljenih orodjih. Sledi sistematični popis vseh aktivnosti in orodij ter izbiranje najustrežnejših. Pri tem se nismo osredotočili le na aktivnosti in orodja, ki zadevajo izključno testiranja mobilnih aplikacij, ampak smo želeli zajeti večino faz procesa, kot jih opredeli RUP [4]. Kljub temu smo pri popisu dali



Slika 3.3: Ogrodje za ocenjevanje v treh korakih.

večji poudarek aktivnostim in orodjem, ki se uporabljajo pri testiranju.

V drugem koraku pripravimo vprašalnike, na katere odgovarjajo ključni deležniki. V vprašalnikih so zajete izbrane aktivnosti in orodja. Vprašalniki so sestavljeni kot trditve, ki izhajajo iz karakteristik posameznega vidika, ki smo jih obravnavali v Poglavju 3.2. Za vrednotenje trditev smo uporabili sedemstopenjsko Likertovo lestvico (ang. *Likert scale*), ki se uporablja za vrednotenje sklopa trditev [74], z naslednjo mersko lestvico:

1	močno se ne strinjam
2	se ne strinjam
3	delno se ne strinjam
4	nevtravno
5	delno se strinjam
6	se strinjam
7	močno se strinjam

Prvi vprašalnik v Tabeli 3.1 je namenjen vodstvu in upravi podjetja, ter služi ocenjevanju ekonomskega vpliva posameznih trditev. Drugi vprašalnik v Tabeli 3.2 je namenjen tehničnim vodjem, ki ocenjujejo tehnično ustreznost posameznih trditev, kar pokriva tehnični vidik. Tretji vprašalnik v Tabeli 3.3 je sestavljen za celotno razvojno skupino in z njim se ocenjuje uporabniška sprejetost oz. sociološki vidik. Pri tem vprašalniku anketiranec vrednoti le aktivnosti in orodja, ki jih sam aktivno izvaja oz. uporablja.

V tretjem koraku izvedemo vprašalnike s ključnimi deležniki. Pri tem je deležnike treba motivirati, da bodo rezultati vprašalnikov pomagali odkriti pomanjkljivosti pri procesu in bodo pozitivno vplivali na njihovo delo. Nato je opravljena analiza rezultatov, ki izpostavi aktivnosti in orodja, ki so bila ocenjena pod povprečjem. Za takšne aktivnosti in orodja se nato pripravi sezname ukrepov za izboljšanje, ki se predstavijo vodstvu in tehničnim vodjem podjetja.

Ekonomski vidik	
Čiste koristi	Izvajanje <aktivnosti> v veliki meri poveča čiste koristi za naše podjetje.
	Uporaba <orodja> v veliki meri poveča čiste koristi za naše podjetje.
Zadovoljstvo kupcev	Izvajanje <aktivnosti> izpolnjuje naročnikove/kupčeve zahteve.
	Uporaba <orodja> izpolnjuje naročnikove/kupčeve zahteve.
Strokovno učenje	Izvajanje <aktivnosti> zaposlenim omogoča pridobivanje novih znanj.
	Uporaba <orodja> zaposlenim omogoča pridobivanje novih znanj.

Tabela 3.1: Vzorec vprašalnika s trditvami, s katerimi se je ocenjeval ekonomski vidik.

Tehnični vidik	
Skladnost s standardi informacijskih tehnologij	<Aktivnost> je skladna z informacijskimi tehnologijami in notranjimi standardi organizacije.
	Uporaba <orodja> je skladna z informacijskimi tehnologijami in notranjimi standardi organizacije.
Vpliv na organizacijo	Izvajanje <aktivnosti> omogoča uresničitev strateških ciljev podjetja.
	Uporaba <orodja> omogoča uresničitev strateških ciljev podjetja.
Skladnost s tehničnimi potrebami projekta	<Aktivnost> je skladna s tehničnimi potrebami in zahtevami podjetja.
	Uporaba <orodja> je skladna s tehničnimi potrebami in zahtevami podjetja.

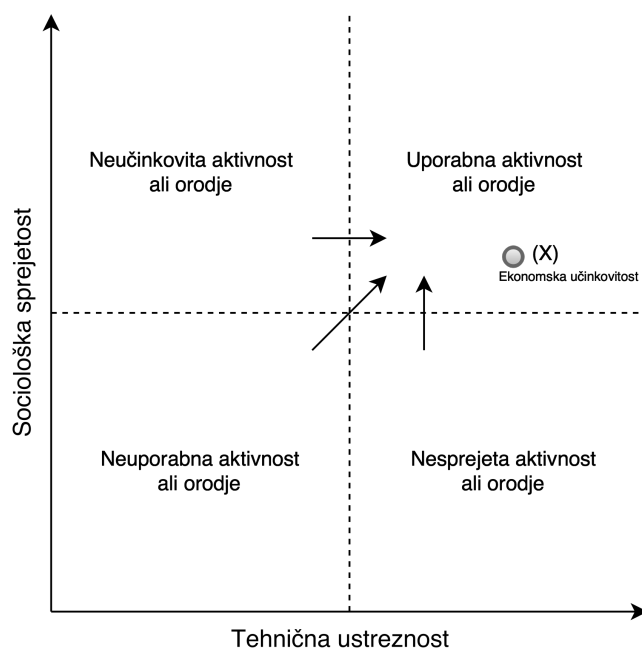
Tabela 3.2: Vzorec vprašalnika s trditvami, s katerimi se je ocenjeval tehnični vidik.

Sociološki vidik	
Pogostost uporabe, če se pojavi priložnost	<Aktivnost> izvajam vedno, ko se pojavi priložnost.
	<Orodje> uporabim vedno, ko se pojavi priložnost.
Relativna prednost	Izvajanje <aktivnosti> mi olajša delo in/ali pohitri delo in/ali izboljša delo.
	Uporaba <orodja> mi olajša delo in/ali pohitri delo in/ali izboljša delo.
Dostopnost do znanja	Imam dostop do vseh virov znanja, ki so potrebna za učinkovito izvajanje <aktivnosti>.
	Imam dostop do vseh virov znanja, ki so potrebna za učinkovito uporabo <orodja>.

Tabela 3.3: Vzorec vprašalnika s trditvami, s katerimi se je ocenjeval sociološki vidik.

3.4 Razvrstitev aktivnosti in orodij

Ogrodje za ocenjevanje se osredotoča na medsebojno interakcijo med sociološkim, tehničnim in ekonomskim vidikom. Pridobljene ocene posameznih aktivnosti in orodij so predstavljene na razsevnem grafikonu, ki je bil predlagan v delu [25]. Grafikon smo prilagodili našim potrebam, saj smo morali predstaviti tri dimenzije na dvodimenzionalnem grafikonu. Posodobljen grafikon prikazovanja ogrodja za ocenjevanje je prikazan na Sliki 3.4. Na grafikonu smo na abscisi (os x) prikazali **tehnično ustreznost**, na ordinati (os y) pa **sociološko sprejetost** aktivnosti ali orodja. Aktivnosti in orodja so tako prikazane kot točke na grafikonu. Zraven vsake točke je prikazana vrednost v oklepaju (X), ki predstavlja tretjo dimenzijo, tj. **ekonomsko učinkovitost**. Glede na povprečno vrednost ekonomske učinkovitosti smo razvrstili aktivnosti in orodja na **ekonomsko neučinkovite aktivnosti ali orodja**, katerih



Slika 3.4: Razsevni diagram, ki vrednotene aktivnosti in orodja razdeli na štiri kategorije. Ekonomski vidik je predstavljen z vrednostjo, zapisano v oklepaju. Po izvedbi predlaganih ukrepov pričakujemo pomaknitev točk v četrti kvadrant.

ekonomski vpliv je pod povprečjem, in na **ekonomsko učinkovite aktivnosti ali orodja**, katerih ekonomski vpliv je nad povprečjem. Grafikon je razdeljen na štiri kvadrante, ki jih določa povprečje vrednosti na obeh oseh. Razdelitev na kvadrante omogoča razvrstitev posameznih aktivnosti in orodij na naslednji način:

- **Neuporabna aktivnost ali orodje** sta sociološko nesprejeta in tehnično neustrezna. Vzrok za neustreznost je lahko posledica nenehnega napredka tehnologije, s čimer lahko določena aktivnost postane zastarela in s tem neustrezna. Posledično se aktivnost ali orodje prenehata uporabljati in tako postaneta popolnoma neuporabna. Neuporabno aktivnost ali orodje je treba odstraniti ali zamenjati z ustreznijšim elementom. Opisane aktivnosti in orodja najdemo v prvem kvadrantu, kar

pomeni, da sta vrednosti sociološke sprejetosti in tehnične ustreznosti pod povprečjem.

- **Neučinkovita aktivnost ali orodje** sta sociološko sprejeta, vendar tehnično neustrezna. To pomeni, da ne ustrezata tehničnim potrebam projekta ali organizacije. Primer so aktivnosti, ki so bile pri prejšnjih projektih tehnično učinkovite in s strani razvijalcev dobro sprejete, vendar pa zaradi napredka tehnologije ali spremembe projekta tehnično ne ustrezajo več trenutnemu projektu. Opisane aktivnosti in orodja najdemo v drugem kvadrantu, kar pomeni, da je vrednost sociološke sprejetosti nad povprečjem, medtem ko je vrednost tehnične ustreznosti pod povprečjem.
- **Nesprejeta aktivnost ali orodje** sta tehnično ustrezna, vendar ne sprejeta z uporabniškega vidika. Vzrok za to je lahko prevelika zahtevnost ali neskladnost s trenutnimi izkušnjami in znanjem ekipe. Opisane aktivnosti in orodja najdemo v tretjem kvadrantu, kar pomeni, da je vrednost tehnične ustreznosti nad povprečjem, medtem ko je vrednost sociološke sprejetosti pod povprečjem.
- **Uporabna aktivnost ali orodje** sta tako sociološko kot tudi tehnično ustrezna. To pomeni, da tehnično zadoščata vsem potrebam projekta ali podjetja in ju vsi uporabniki sprejemajo. Opisane aktivnosti in orodja najdemo v četrtem kvadrantu, kar pomeni, da sta vrednosti sociološke sprejetosti in tehnične ustreznosti nad povprečjem.

Predstavljena razvrstitev omogoča identifikacijo ustreznih in neustreznih aktivnosti in orodij. Osredotočili smo se predvsem na neustrezne aktivnosti in orodja ter predstavili rešitve za njihovo izboljšanje. Rešitve so opredeljene kot seznam konkretnih ukrepov, ki jih je treba izvesti za izboljšanje stopnje sprejetosti, ustreznosti in učinkovitosti določene aktivnosti in orodja. Iskanje rešitev za izboljšanje pa ni odvisno le od procesa vrednotenja posamezne aktivnosti in orodja, temveč so potrebne tudi dodatne informacije, ki jih

pridobimo iz pogovorov z uporabniki metodologije. Čeprav so posamezne aktivnosti in rešitve zanje različne, lahko določimo splošne ukrepe, ki nam bodo pomagali pri izboljšanju [25]:

- V primeru **neuporabne** aktivnosti ali orodja, ki sta sociološko nesprejeta, tehnično neustrezna in ekonomsko neučinkovita, je logičen ukrep, da takšno aktivnost ali orodje zamenjamo ali v celoti zavržemo. V takšnem primeru je možno, da aktivnost ali orodje sploh nista potrebna. Skoraj zagotovo lahko najdemo aktivnost ali orodje, ki sta tehnično, sociološko in ekonomsko ustrežnejša.
- V primeru **neučinkovite** aktivnosti ali orodja moramo izboljšati tehnično ustreznost ter pri tem ohraniti sociološko sprejetost in ekonomsko učinkovitost. Ker so uporabniki metodologije že sprejeli aktivnost ali orodje, si je treba prizadevati, da se aktivnost ali orodje spremenita do te mere, da ponovno postaneta tehnično ustrezna.
- V primeru **nesprejete** aktivnosti ali orodja moramo najti vzroke, ki povzročajo zavračanje aktivnosti ali orodja med njunimi uporabniki. Razlog za to je lahko pomanjkanje znanja in izkušenj. Za povečanje sprejetosti lahko v tem primeru dodatno izobrazimo uporabnike ter jim predstavimo prednosti izvajanja aktivnosti ali uporabe orodja.
- Aktivnosti in orodja, ki se izkažejo za **ekonomsko učinkovite**, vendar pa tehnično niso ustrezni in/ali so nesprejeti, je potrebno treba prioritarno, saj imajo večji vpliv na uspešnost podjetja kot tisti, ki so **ekonomsko neučinkoviti**.

Pričakovali smo, da se bodo po apliciranju rešitev za izboljšanje aktivnosti in orodja premaknili v četrti kvadrant, v katerem se nahajajo uporabne aktivnosti in orodja. Vendar to še ne pomeni, da pri posameznih aktivnostih in orodjih ni prostora za izboljšave. Treba je tudi omeniti, da je spreminjanje metodologije dolgotrajen in zahteven proces.

Poglavje 4

Študija primera

V tem poglavju smo najprej predstavili metodologijo, ki smo jo uporabili za preizkus našega modela, in sicer študijo primera. Opisali smo tudi uporabljena orodja, ki smo jih uporabili za namen izvajanja študije primera. Nato smo predstavili tri izbrana podjetja, ki se ukvarjajo z razvojem mobilnih aplikacij, v katerih smo izvedli študijo primera. Na koncu poglavja smo predstavili rezultate študije primera.

4.1 Opis metodologije

Namen študije primera je bil preizkusiti razvito teoretično ogrodje v praksi in dokazati njegovo koristnost. Študija primera je poglobljeno empirično preiskovanje pojava v realnem času znotraj njegovega konteksta, pri čemer meje med pojavom in njegovim kontekstom niso popolnoma očitne [75]. Preiskovanje s študijo primera [75]:

- Se sooča s situacijo, v kateri obstaja veliko več spremenljivk, kot je na voljo podatkov.
- Se opira na različne vire dokazov, podatki pa morajo sovpadati.
- Izkorišča prehodni razvoj teoretičnih predlogov, ki vodijo zbiranje in analizo podatkov.

Iz zgornje definicije je možno razbrati, da je študija primera celovita raziskovalna metoda, saj pokriva načrtovanje raziskave, tehniko zbiranja podatkov in pristope k analizi podatkov. Za empirično metodo raziskovanja smo izbrali študijo primera, ker najbolje preučuje kompleksno interakcijo med različnimi deležniki in projekti razvoja informacijskega sistema [75]. Prav tako je najprimernejši način preučevanja dogodkov v realnem okolju, kjer raziskovalci nimajo nadzora nad potekom dogodkov [75].

Študija primera lahko zajema tako **kvalitativne** kot tudi **kvantitativne** metode raziskovanja. Pri izvajanju študije primera v izbranih podjetjih smo uporabili oba pristopa raziskovanja. Kvalitativne metode smo uporabili pri izvedbi intervjujev in pogovorov z vodstvom in tehničnimi vodji za identifikacijo metodologije razvoja in testiranja mobilnih aplikacij ter kasneje za preverjanje koristnosti našega modela. Kvantitativne metode raziskovanja smo uporabili pri pripravi vprašalnikov in statistični obdelavi podatkov. Za postavitev vprašalnikov smo uporabili spletno orodje 1ka.si [76], ki omogoča izvoz rezultatov v poljubnem formatu. Zbrane podatke iz vprašalnikov smo analizirali s pomočjo programa IBM SPSS Statistics [77], ki med drugim omogoča izris razsevnih diagramov (ang. *scatter plot*) in škatel z brki (ang. *box plot*).

Za izvajanje študije primerov smo izbrali tri podobna podjetja. Vsa tri podjetja so majhna slovenska podjetja, ustanovljena v obdobju zadnjih šestih let, ki se primarno ukvarjajo z razvojem mobilnih aplikacij. Ker so razvojne ekipe v izbranih podjetjih majhne, razvijalci poleg razvijanja v veliki meri izvajajo tudi aktivnosti testiranja. Podjetja, v katerih smo izvedli študijo primera, smo podrobneje opisali v naslednjem Poglavju 4.2. Študijo primera smo izvedli v treh podjetjih, da lahko s pomočjo replikacije rezultatov [75] bolje ocenimo koristnost našega predstavljenega ogrodja. S tem, ko smo v študijah primera referenčnih podjetij ocenili koristnost predlaganega ogrodja, smo lahko odgovorili na vprašanje o koristnosti našega ogrodja.

Pri izvajanju študije smo sledili smernicam najboljših praks na tem področju, zato smo študijo primera izvedli v naslednjih fazah [75]:

- **Načrtovanje študije primera** – Razvoj teorije, propozicij in ključnih izzivov v predvideni študiji. Opredelijo se postopki za zagotavljanje kakovosti študije primera.
- **Priprava na zbiranje podatkov** – Seznanjanje z izbranim podjetjem za raziskavo in razvoj protokola študije primera.
- **Zbiranje podatkov** – Sledenje protokolu študije primera. Uporabijo se različni viri podatkov in izdelajo se baze podatkov o študiji primera.
- **Analiza dokazov** – Uporaba analitičnih metod z uporabo kvalitativnih in kvantitativnih podatkov.
- **Poročanje o rezultatih** – Izdelava besedil in vizualnih dokazov.

4.2 Študija primera na izbranih podjetjih

V tem poglavju smo podrobneje opisali podjetja, v katerih smo izvedli študijo primera. Kot smo že omenili, smo izbrali tri podobna podjetja. Podobna so si v tem, da so majhna in mlajša slovenska podjetja, ki se ukvarjajo z razvojem mobilnih aplikacij.

4.2.1 Podjetje 1

Prva študija primera je bila izvedena v manjšem, novoustanovljenem slovenskem podjetju. Podjetje se ukvarja z izdelavo tipskih mobilnih aplikacij za platformi Android in iOS, s sistemi za upravljanje vsebin in drugimi digitalnimi orodji, ki naročnikom povečujejo konkurenčno prednost na trgu. V podjetju je zaposlenih okoli deset ljudi, od tega je večina razvijalcev, imajo pa tudi dva inženirja kakovosti. Pri izvajanju študije primera smo se osredotočili le na del procesa, ki zadeva razvoj in testiranje mobilnih aplikacij. Podjetje ima lastno razvito ohlapno metodologijo razvoja mobilnih aplikacij, ki predpisuje uporabo določenih aktivnosti, postopkov in orodij. Metodologija ni točno formalizirana, ampak se odraža v obliki dogovorov in smernic,

ki jih predlaga direktor podjetja skupaj z vodjema razvijalcev. Zaradi neformalizacije je metodologija prilagodljiva.

V podjetju smo identificirali tri ključne deležnike, in sicer direktorja, tehnična vodja ter člane razvojne ekipe, v katero sodijo tako razvijalci kot tudi inženirja kakovosti. V študiji primera je sodelovalo šest članov razvojne ekipe. Delo na projektih po navadi organizirajo v dve razvojni ekipi, ki jih vodita tehnična vodja. Od nivojev testiranja, ki smo jih predstavili v Poglavju 2.3.2, se podjetje najbolj osredotoča na integracijsko in sistemsko testiranje.

V Tabeli 4.2.1 so predstavljene izbrane aktivnosti in orodja, ki se v podjetju uporabljajo pri razvoju in testiranju mobilnih aplikacij. Vseh 19 aktivnosti skupaj predstavlja metodologijo, ki jo uporablja podjetje, ko izdeluje in testira mobilne aplikacije. Pri tem člani razvojne ekipe ne opravljajo vseh aktivnosti v tabeli. Nekateri razvijalci razvijajo le mobilne aplikacije za določeno platformo, nekateri za obe, nekateri pa zaledne sisteme. Razvijalci opravljajo tudi večino aktivnosti in uporabljajo orodja, ki zadevajo testiranje. Inženirja kakovosti po večini opravljata le aktivnosti, ki so povezane s testiranjem.

Tabela 4.2.1

Seznam vseh ocenjenih aktivnosti in orodij v Podjetju 1

Q1 Zajem in analiza zahtev

Q2 Priprava tehnične specifikacije

Q3 Uporaba orodja Asana za beleženje napredka in hroščev

Q4 Priprava testnega načrta

Q5 Razvoj aplikacije iOS v programu Xcode

Q6 Testiranje mobilne aplikacije v programu Xcode

Q7 Razvoj aplikacije Android v programu Android Studio

Q8 Uporaba emulatorja Genymotion za testiranje

Q9 Avtomatsko testiranje z orodjem Monkey

Q10 Preverjanje delovanja aplikacije s programom Monitor

- Q11 Notranji tedenski sestanki*
- Q12 Testiranje funkcionalnosti mobilne aplikacije*
- Q13 Testiranje uporabnosti mobilne aplikacije*
- Q14 Testiranje uporabniškega vmesnika*
- Q15 Regresijsko testiranje*
- Q16 Razvoj zalednega sistema v programu Eclipse*
- Q17 Testiranje potisnih sporočil*
- Q18 Uporaba orodja Advanced Rest Client za testiranje klicev REST*
- Q19 Testiranje dogodkov v bazi Big Query*

4.2.2 Podjetje 2

Druga študija primera je bila izvedena v manjšem podjetju, ki se ukvarja izključno z razvojem mobilnih iger. Ekipa je bila sprva del večjega podjetja, v katerem so razvijali igre kot del notranjega projekta, nato pa je bilo ustanovljeno novo podjetje. Podjetje razvija mobilne igre za platformi Android in iOS. V podjetju je zaposlenih do deset ljudi, med katerimi je največ zaposlenih razvijalcev, imajo pa tudi inženirje kakovosti. Podjetje ima lastno razvito metodologijo razvoja mobilnih aplikacij, ki pa ni točno formalizirana. Metodologija predpisuje uporabo določenih postopkov, aktivnosti in orodij pri razvoju mobilnih iger, ki se spreminja in prilagaja po potrebi.

V podjetju smo opredelili enake vloge kot v Podjetju 1, in sicer direktorja, tehničnega vodja ter razvojno ekipo. V študiji primera so sodelovali štirje člani razvojne ekipe. Testiranje izvajajo tako razvijalci kot tudi inženirji kakovosti. Razvojna ekipa pri testiranju največ poudarka nameni sistemskemu testiranju, ki smo ga predstavili v Poglavju 2.3.2. Podjetje se zaveda tudi pomembnosti uporabniške izkušnje in uporabnosti mobilnih iger.

V Tabeli 4.2.2 so predstavljene izbrane aktivnosti in orodja, ki se v podjetju uporabljajo pri razvoju in testiranju mobilnih iger. Seznam aktivnosti in orodij skupaj predstavlja neformalno metodologijo, ki določa proces razvoja mobilnih iger. Posamezni član razvojne ekipe ne izvaja vseh navedenih aktivnosti in ne uporablja vseh navedenih orodij. Razvijalci poleg aktivnosti

in orodij, ki so za njih značilna, izvajajo tudi nekatere aktivnosti testiranja. Inženirji kakovosti običajno opravljajo le aktivnosti, ki se tičejo testiranja.

Tabela 4.2.2

Seznam vseh ocenjenih aktivnosti in orodij v Podjetju 2

- Q1 Iskanje idej in konceptov*
- Q2 Priprava tehnične specifikacije*
- Q3 Izdelava prototipa*
- Q4 Uporaba orodja Trello za beleženje napredka*
- Q5 Priprava grafičnih modelov*
- Q6 Uporaba orodja Jira za odpiranje novih težav (hroščev)*
- Q7 Razvoj mobilne igre v programu Unity*
- Q8 Uporaba orodja Bitbucket pri razvoju*
- Q9 Priprava testnega načrta*
- Q10 Testiranje uporabniškega vmesnika s simulatorjem Unity*
- Q11 Uporaba orodja Unity Cloud Build za grajenje aplikacij*
- Q12 Uporaba orodja Crashlytics za iskanje napak*
- Q13 Uporaba orodja Swrve za testiranje*
- Q14 Testiranje uporabniške izkušnje*
- Q15 Testiranje funkcionalnosti mobilne aplikacije*
- Q16 Testiranje uporabnosti mobilne aplikacije*
- Q17 Regresijsko testiranje*
- Q18 Notranji sestanki*

4.2.3 Podjetje 3

Tretja in zadnja študija primera je bila izvedena v manjšem podjetju, ki je bilo ustanovljeno leta 2011. Podjetje naročnikom nudi celovite programske rešitve, katerih del so tudi mobilne rešitve. Mobilne aplikacije razvijajo za platformi Android in iOS. V podjetju je zaposlenih okoli petnajst ljudi. Večina zaposlenih je del razvojnih ekip, med njimi pa ni inženirja kakovosti. Podjetje ima lastno razvito metodologijo razvoja programskih rešitev,

ki je zelo ohlapno opredeljena. Metodologija opisuje smernice za razvojno ekipo, kot na primer dobre prakse programiranja ter aktivnosti in orodja, ki se uporabljajo pri razvoju programskih rešitev. Metodologija je zaradi svoje ohlapnosti prilagodljiva različnim vrstam projektov. Ker podjetje nudi raznovrstne programske rešitve, smo se osredotočili le na vrednotenje aktivnosti in orodij, ki se uporabljajo pri razvoju in testiranju mobilnih aplikacij.

V podjetju smo opredelili tri vloge, in sicer direktorja, tehničnega vodja ter člane razvojne ekipe. V študijo primera je bilo vključenih šest članov razvojne ekipe. Podjetje dela za več naročnikov naenkrat, kar pomeni, da se razdelijo v manjše razvojne ekipe, na čelu katerih je tehnični vodja. Podjetje ima dva tehnična vodja, ki vodita razvojne ekipe in nadzirata potek razvoja. Razvojne ekipe lahko skupaj s tehničnim vodjem hkrati delajo na več projektih. Ker podjetje nima zaposlenega inženirja kakovosti, večino testiranja opravijo razvijalci sami. Od nivojev testiranja, ki smo jih predstavili v Poglavju 2.3.2, se najbolj osredotočajo na sistemsko testiranje. Velik poudarek dajo na uporabnost aplikacije in na dobro uporabniško izkušnjo.

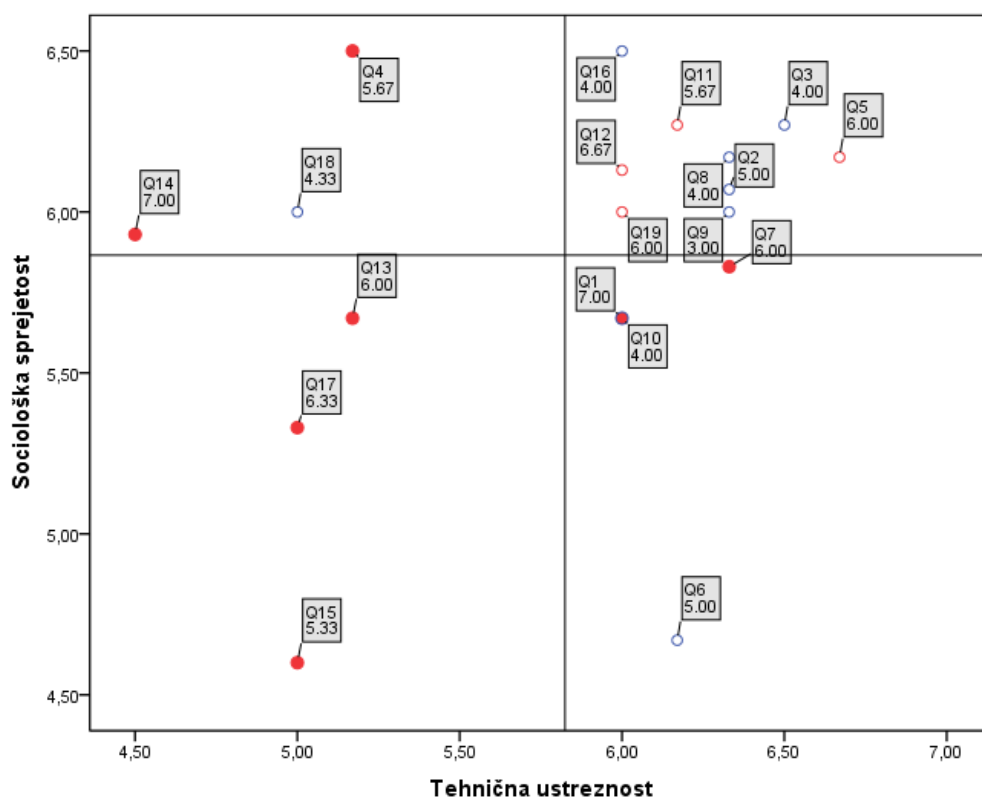
V Tabeli 4.2.3 je predstavljen seznam izbranih aktivnosti in orodij, ki se v podjetju uporabljajo pri razvoju in testiranju mobilnih aplikacij. Popisane aktivnosti in orodja skupaj predstavljajo metodologijo, ki zajema 19 obveznih elementov, ki določajo izvajanje aktivnosti in uporabo orodij. Pri tem posamezen član razvojne ekipe ne izvaja vseh aktivnosti v tabeli, saj je v razvojni ekipi več vrst razvijalcev. Razlikujejo se predvsem po platformi, za katero razvijajo, nekateri pa razvijajo izključno zaledne sisteme. Pri tem pa vsak razvijalec opravlja tudi večino aktivnosti, ki so povezane s testiranjem.

Tabela 4.2.3*Seznam vseh ocenjenih aktivnosti in orodij v Podjetju 3*

- Q1 Zajem zahtev*
- Q2 Priprava tehnične specifikacije*
- Q3 Izdelava osnutkov ali prototipov*
- Q4 Uporaba orodja Jira za beleženje napredka po nalogah/hroščih*
- Q5 Priprava testnega načrta*
- Q6 Priprava vprašalnika Q&A*
- Q7 Razvoj aplikacije iOS v programu Xcode*
- Q8 Testiranje mobilne aplikacije v programu Xcode*
- Q9 Razvoj aplikacije Android v programu Android Studio*
- Q10 Testiranje mobilne aplikacije z emulatorjem Android*
- Q11 Uporaba orodja Git za verzioniranje*
- Q12 Uporaba orodja Jenkins za pregledovanje kode*
- Q13 Testiranje funkcionalnosti mobilne aplikacije*
- Q14 Testiranje uporabnosti mobilne aplikacije*
- Q15 Testiranje uporabniške izkušnje*
- Q16 Testiranje zmogljivosti aplikacije*
- Q17 Razvoj zalednega sistema v programu Netbeans*
- Q18 Testiranje klicev REST*
- Q19 Notranji tedenski sestanki*

4.3 Rezultati študije primera

V tem poglavju smo predstavili rezultate razvitega ogrodja, ki smo ga predstavili v Poglavju 3. Ogrodje smo preverili s študijo primera v izbranih podjetjih, zato smo v nadaljevanju predstavili rezultate vsakega podjetja posebej. Rezultate smo predstavili v obliki razsevnih diagramov, s pomočjo katerih lahko hitro identificiramo podpovprečno ocenjene aktivnosti in orodja. Skupaj z vodstvom podjetij smo nato izbrali kandidate za izboljšanje ter za njih



Slika 4.1: Grafični prikaz ovrednotenih aktivnosti in orodij v Podjetju 1.

podali ukrepe za izboljšanje. Ukrepe smo nato predali vodstvu, ki jih je lahko sprejelo ali zavrnilo.

4.3.1 Podjetje 1

Zaposleni Podjetja 1 so aktivnosti in orodja iz Tabele 4.2.1 ocenili s pomočjo razvitega ogrodja, ki smo ga predstavili v Poglavlju 3. Rezultati ocenjevanja posameznih aktivnosti in orodij so prikazani v Tabeli 4.1. Vsi ključni deležniki so izpolnili vprašalnike. Tehnična vodja sta ocenjevala tehnični vidik, člani razvojnih ekip so ocenjevali sociološki vidik, direktor pa je ocenjeval ekonomski vidik.

Rezultati vrednotenja aktivnosti in orodij s strani vseh ključnih deležnikov

	Tehnični vidik				Sociološki vidik				Ekonomski vidik			
E	T1	T2	T3	PT	S1	S2	S3	PS	E1	E2	E3	PE
Q1	6	6.5	5.5	6	5.8	5.2	6	5.67	7	7	7	7
Q2	6.5	6.5	6	6.33	6.2	6	6	6.07	5	5	5	5
Q3	6.5	6.5	6.5	6.5	6	6.2	6.6	6.27	4	4	4	4
Q4	5	5.5	5	5.17	6.25	6.25	7	6.5	6	6	5	5.67
Q5	6.5	7	6.5	6.67	6	6	6.5	6.17	6	6	6	6
Q6	6	6	6.5	6.17	4.5	5.5	4	4.67	5	5	5	5
Q7	6.5	6.5	6	6.33	7	7	3.5	5.83	6	6	6	6
Q8	7	6	6	6.33	5.5	6.5	6.5	6.17	4	4	4	4
Q9	6.5	6	6.5	6.33	5.4	6.2	6.4	6	3	3	3	3
Q10	6	5.5	6.5	6	5.33	5.33	6.33	5.67	4	4	4	4
Q11	6.5	6	6	6.17	6	6.2	6.6	6.27	6	6	5	5.67
Q12	6	6	6	6	6.6	6	5.8	6.13	7	7	6	6.67
Q13	5.5	5	5	5.17	6	5.6	5.4	5.67	6	6	6	6
Q14	4	4.5	5	4.5	5.6	6.6	5.60	5.93	7	7	7	7
Q15	5	4.5	5.5	5	4.8	2.8	6.2	4.6	5	6	5	5.33
Q16	5.5	6.5	6	6	7	6.5	6	6.5	4	4	4	4
Q17	5	5	5	5	6	4.67	5.33	5.33	6	7	6	6.33
Q18	5.5	5	4.5	5	5.5	6.25	6.25	6	4	5	4	4.33
Q19	6	6	6	6	6.5	6	5.5	6	6	6	6	6
P.				5.82				5.86				5.32

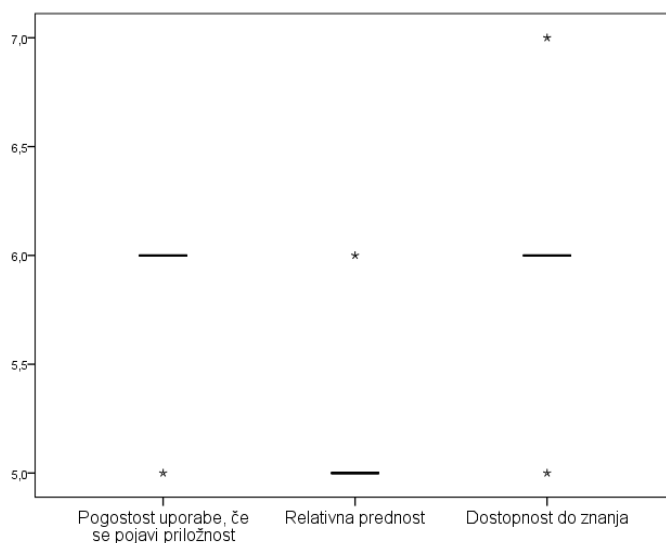
Tabela 4.1: Rezultati ocenjevanja posameznih aktivnosti in orodij v Podjetju 1. Stolpec **E** označuje aktivnosti in orodja po Tabeli 4.2.1. Stolpci **T1**, **T2** in **T3** predstavljajo povprečno vrednost karakteristik tehničnega vidika: skladnost s standardi informacijskih tehnologij, vpliv na organizacijo in skladnost s tehničnimi potrebami podjetja. Stolpci **S1**, **S2** in **S3** predstavljajo povprečno vrednost karakteristik sociološkega vidika: pogostost uporabe, če se pojavi priložnost, relativna prednost in dostopnost do znanja. Stolpci **E1**, **E2** in **E3** predstavljajo povprečno vrednost karakteristik ekonomskega vidika: čiste koristi, zadovoljstvo kupcev, strokovno učenje. Stolpci **TP**, **SP** in **EP** predstavljajo povprečno vrednost vseh karakteristik danega vidika za posamezno aktivnost ali orodje.

so vidni na Sliki 4.1. Kot je razvidno iz razsevnega diagrama, se je veliko aktivnosti uvrstilo v četrti diagram. To pomeni, da so aktivnosti in orodja sociološko sprejeta ter tehnično ustrezna. Nekaj pa je tudi takih, ki se niso uvrstila v četrti kvadrant. Takšne aktivnosti in orodja so sociološko nesprejeta ali tehnično neustrezna ali pa kar oboje ter so kandidati za izboljšanje. Z vodstvom smo se dogovorili, da je v interesu podjetja izboljšati tiste aktivnosti in orodja, ki se niso uvrstila v četrti kvadrant in imajo visoko oceno z ekonomskega vidika. Visoka ocena ekonomske dimenzije namreč pomeni, da imata izbrana aktivnost ali orodje visok vpliv na vsesplošno uspešnost podjetja. Aktivnosti in orodja s podpovprečnim ekonomskim vplivom so na razsevnem diagramu označena z modro barvo, medtem ko so tista z nadpovprečnim ekonomskim vplivom označena z rdečo barvo. Točna vrednost ekonomskega vpliva je zapisana v kvadratu poleg točke.

V nadaljevanju smo predstavili izbrane aktivnosti in orodja, ki smo jih poskusili izboljšati. Ta morajo ustrezati zgornjim pogojem, torej so podpovprečno ocenjena z vsaj enega vidika in imajo visok ekonomski vpliv. Kandidati za izboljšanje so na razsevnem diagramu predstavljeni z rdečim polnilom.

- **(Q1) Zajem in analiza zahtev**

Aktivnost zajem in analiza zahtev se v podjetju izvaja na začetku vsakega razvoja mobilne aplikacije. Podjetje se dobi z naročnikom, popiše njegove zahteve in jih kasneje analizira. Aktivnost je bila ocenjena kot tehnično ustrezna, vendar sociološko nesprejeta. Zaradi visoke ocene vpliva na uspešnost podjetja je aktivnost kandidat za izboljšanje. Na Sliki 4.2 je prikazana razporeditev ocen sociološkega vidika. Vidimo, da ima razvojna ekipa podobno mnenje, saj pri vsaki karakteristiki prevladuje enaka ocena, če odstranimo osamelce. Najslabše je ocenjena karakteristika relativna prednost. Razlog za slabšo oceno je v tem, da se zahteve tekom razvoja velikokrat spremenijo. Za razvijalce to pomeni veliko dvojnega dela, spremenijo pa se tudi časovne ocene ključka projekta. Vodstvu smo predlagali, naj se analiza zahtev izvaja sistematičneje in po principu minimalnega izvedljivega izdelka (ang.

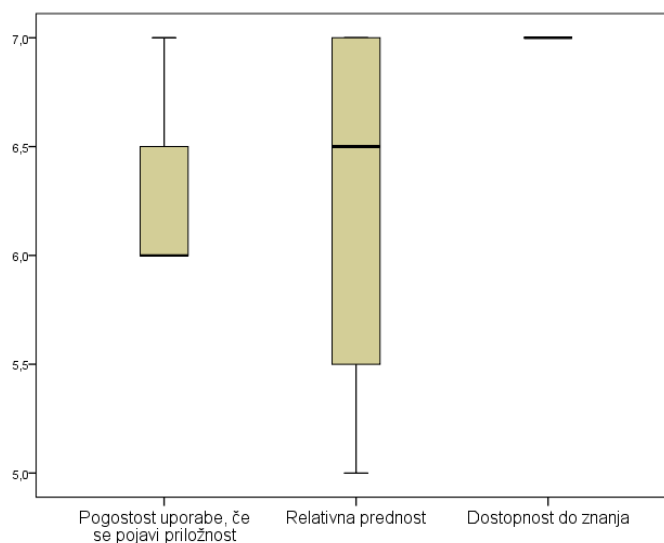


Slika 4.2: Prikaz razporeditve ocen sociološkega vidika za aktivnost Q1.

Minimum Viable Product – MVP). Naročnik naj tako točneje opredeli prednost zelenih funkcionalnosti, ki se lahko razvijejo v določenem časovnem in stroškovnem okviru. Z ukrepom bi pozitivno vplivali na sprejetost s strani uporabnikov, ostali dimenziji pa bi ostali nespremenjeni. Aktivnost bi se tako prestavila v četrti kvadrant, v katerem so uporabne aktivnosti in elementi. Predlog je vodstvo sprejelo.

- **(Q4) Priprava testnega načrta**

Aktivnost priprava testnega načrta se pripravi po zajemu in analizi zahtev. Testni načrt se pripravi za vsako funkcionalnost in je sestavljen iz testnih scenarijev in primerov. Na Sliki 4.3 je prikazana razporeditev ocen sociološkega vidika, ki je nadpovprečna, torej je sprejeta s strani uporabnikov, vendar ima nizko oceno tehnične primernosti. Trenutno se testni scenariji in primeri pripravljajo v dokumentih ali pa v razporednicah, kar za tehnična vodja ni dovolj skladno z informacijskimi tehnologijami in tehničnimi potrebami podjetja. Da bi povečali tehnično ustreznost smo direktorju in tehničnima vodjema predlagali orodje TestRail [81]. Orodje poveča učinkovitost pri pisanju testnih scenarijev

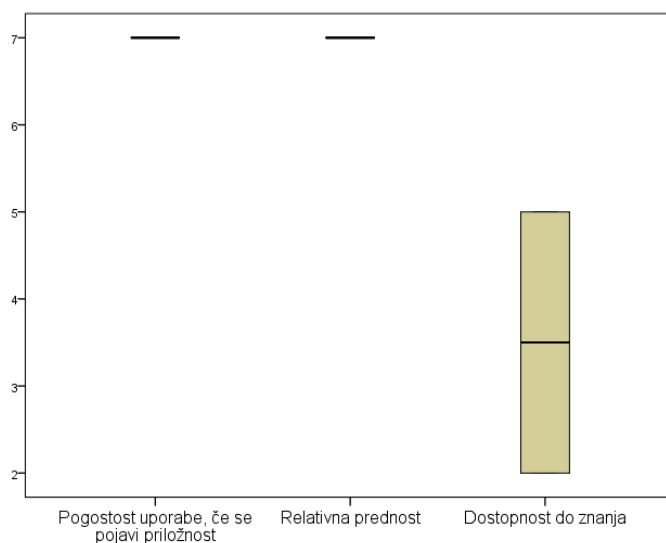


Slika 4.3: Prikaz razporeditve ocen sociološkega vidika za aktivnost Q4.

in primerov, poleg tega pa nudi izdelavo poročil ter prikazuje časovni potek testiranja. Z uporabo tehnično bolj dovršenega orodja bi se aktivnost prestavila v četrti kvadrant. Nakup licenc za orodje bi pomenil negativen vpliv na ekonomsko komponento. Predlog smo predstavili vodstvu in tehničnima vodjema. Predlog je bil deloma sprejet – strinjajo se, da je treba začeti uporabljati tehnično bolj dovršeno orodje, vendar bodo raje poiskali orodje, ki je brezplačno in tako ne bo imelo negativnega vpliva na ekonomsko komponento.

- **(Q7) Razvoj aplikacije Android v programu Android Studio**

Razvoj mobilne aplikacije za platformo Android se izvaja v programu Android Studio. Aktivnost je tehnično ustrezna, ima visoko oceno ekonomske komponente, vendar je njena sociološka ocena malo pod povprečjem. Iz Slike 4.4 lahko razberemo, da pri ocenah karakteristik sociološkega vidika pogostost uporabe, če se pojavi priložnost, ter relativna prednost ni odstopanja, saj so ju razvijalci, ki razvijajo aplikacije Android, ocenili z najboljšo možno oceno. Večje odstopanje je pri karakteristiki dostopnost do znanja, pri kateri ocene variirajo ven-

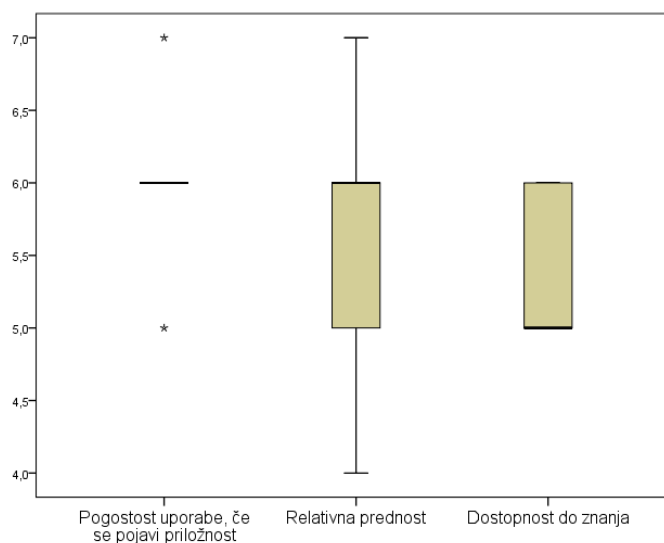


Slika 4.4: Prikaz razporeditve ocen sociološkega vidika za aktivnost Q7.

dar je razvidno, da je ta karakteristika ocenjena zelo nizko. Zaposleni menijo, da nimajo vsega znanja, ki ga potrebujejo za razvoj mobilnih aplikacij za platformo Android in menijo, da se lahko še marsikaj naučijo. Vodstvu in tehničnima vodjema smo predlagali, naj poskrbijo za izobraževanje ter razvijalce prijavijo na delavnice ali forume, kjer se lahko naučijo veliko novega. Z ukrepom bi zvišali tretjo karakteristiko sociološkega vidika, medtem ko bi tehnična ocena ostala nespremenjena. Ukrep bi vplival tudi na oceno ekonomske komponente, saj izobraževanje predstavlja strošek, vendar bi se ta vpliv izničil, saj bi zaposleni pridobili nova znanja, kar bi pozitivno vplivalo na uspešnost podjetja. Direktor in tehnična vodja so predlog sprejeli. Skupaj z razvijalci bodo skušali najti najprimernejšo delavnico, ki se je bodo udeležili in tako razširili svoje znanje programiranja.

- **(Q13) Testiranje uporabnosti mobilne aplikacije**

Testiranje uporabnosti mobilne aplikacije se v podjetju izvaja med ra-



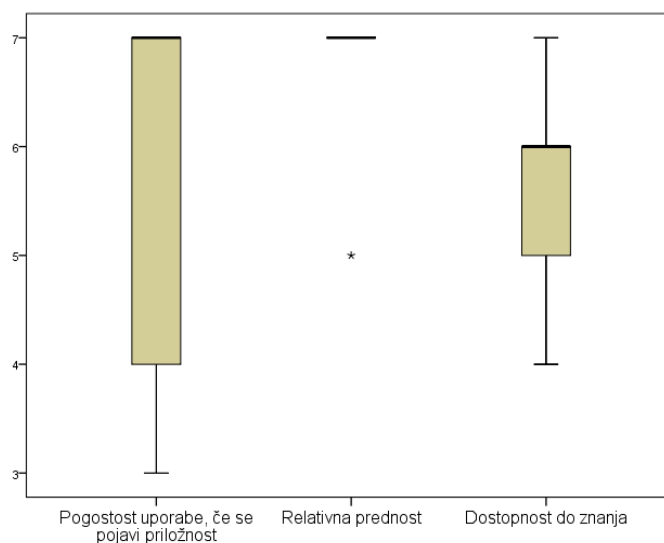
Slika 4.5: Prikaz razporeditve ocen sociološkega vidika za aktivnost Q13.

zvojem mobilne aplikacije za preverjanje uporabnosti posameznih funkcionalnosti. Izvajajo se testi na uporabnikih, ki se odvijajo v obliki sestankov v eni izmed pisarn. Oseba, ki vodi izvajanje uporabniških testov, si ugotovitve zapisuje na list papirja. Prav zaradi staromodnega pisanja na papir je bila aktivnost ocenjena kot tehnično neustrezna. Za povečanje tehnične ustreznosti smo predlagali, da se list papirja zamenja s predlogo vprašalnika v računalniški obliki. Ker je pri testiranju uporabnosti treba spremljati uporabnikove reakcije, smo predlagali tudi nakup kamere za snemanje uporabniških testov. S posnetki se namreč lažje analizira obnašanje in občutke uporabnikov pri uporabi mobilne aplikacije. Aktivnost se nahaja v prvem kvadrantu, torej ima poleg podpovprečne ocene iz tehnične ustreznosti tudi podpovprečno oceno iz uporabniške sprejetosti. Razporeditev ocen posameznih socioloških karakteristik je predstavljena na Sliki 4.5, na kateri lahko vidimo, da je večina članov razvojne ekipe za karakteristiko pogostost uporabe, če se pojavi priložnost, podala nadpovprečno oceno šest. Pri karakteristiki relativna prednost je malce več odstopanja,

saj je prejela več različnih ocen. Pri ocenjevanju karakteristike dostopnost do znanja so imeli uporabniki podobno mnenje in jo ocenili podpovprečno. Oceno sociološkega vidika bi podjetje lahko povečalo z dodatnim izobraževanjem kadra o opravljanju uporabniških testov in testiranju uporabnosti mobilnih aplikacij. Z upoštevanjem vseh podanih ukrepov bi se aktivnost prestavila v četrti kvadrant in tako postala uporabna. Direktor in tehnična vodja so sprejeli le prvi ukrep, in sicer poskus povečanja tehnične ustreznosti s posodobitvijo tehnologije pri izvajanju uporabniških testov.

- **(Q14) Testiranje uporabniškega vmesnika**

Aktivnost testiranje uporabniškega vmesnika v podjetju predstavlja preverjanje pravilnosti lokalizacij, internacionalizacij, ustreznosti pisave in postavitve grafičnih elementov v različnih dialogih in scenah mobilne aplikacije. Testiranje največkrat izvajata inženirja kakovosti, ki aktivnosti v celoti izvajata ročno z nalaganjem aplikacije na različne mobilne naprave, brez kakršne koli avtomatizacije. Zaradi tega sta tehnična vodja aktivnost ocenila kot tehnično najbolj neustrezno. Aktivnost ima nadpovprečno oceno sociološke sprejetosti, razlog za to pa je dodelan proces ročnega testiranja uporabniških vmesnikov. Razporeditev ocen posameznih karakteristik je predstavljena na Sliki 4.6. Največ odstopanja je pri karakteristiki pogostost uporabe, če se pojavi priložnost za uporabo, pri kateri sta inženirja kakovosti podala najvišjo oceno, medtem ko so razvijalci podali nižje ocene. Razlog za to je, da razvijalci testirajo uporabniški vmesnik v simulatorjih, vendar, kot kažejo rezultati, bolj poredko. Zaradi visoke ocene vpliva na uspeh podjetja je aktivnost kandidat za izboljšavo. Za izboljšanje tehnične komponente, ki bi aktivnost prestavilo v četrti kvadrant, smo vodstvu in tehničnima vodjema predlagali, da se testiranje uporabniškega vmesnika do neke mere ali pa kar v celoti avtomatizira. Razvite aplikacije so namreč povečini tipske, kar pomeni, da se za različne naročnike uporabi podobne predloge za mobilne aplikacije. Avtomatske teste bi lahko

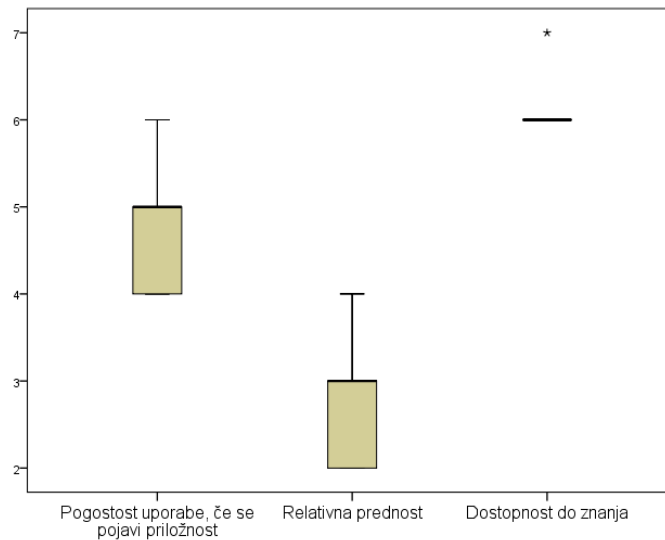


Slika 4.6: Prikaz razporeditve ocen sociološkega vidika za aktivnost Q14.

razvili z ogrođjem Appium [80], ki omogoča avtomatsko testiranje na platformah Android in iOS. Pri tem je tudi možno, da se testi izvajajo na dejanskih mobilnih napravah. To bi zagotovilo, da bi bili testi zanesljivejši in bi identificirali več napak. Na začetku bi se ekonomski vpliv malce zmanjšal, saj razvoj avtomatskih testov zahteva določene vire. Avtomatske teste bi sprva morali vzpostaviti razvijalci, saj imajo več znanja od inženirjev kakovosti, ki nista vešča programiranja. Sčasoma bi se ekonomski vpliv zopet povečal, saj bi bilo testiranje avtomatizirano, kar bi občutno zmanjšalo čas testiranja. Direktor in tehnična vodja so sprejeli ukrep, razvijalci bodo pri razvoju naslednje mobilne aplikacije začeli z razvojem avtomatskih testov.

- **(Q15) Regresijsko testiranje**

Aktivnost regresijsko testiranje se v podjetju izvaja ob nadgradnji različice mobilne aplikacije za zagotavljanje delovanja prejšnjih funkcionalnosti. Mobilne aplikacije se občasno nadgrajuje zaradi optimizacije, od-



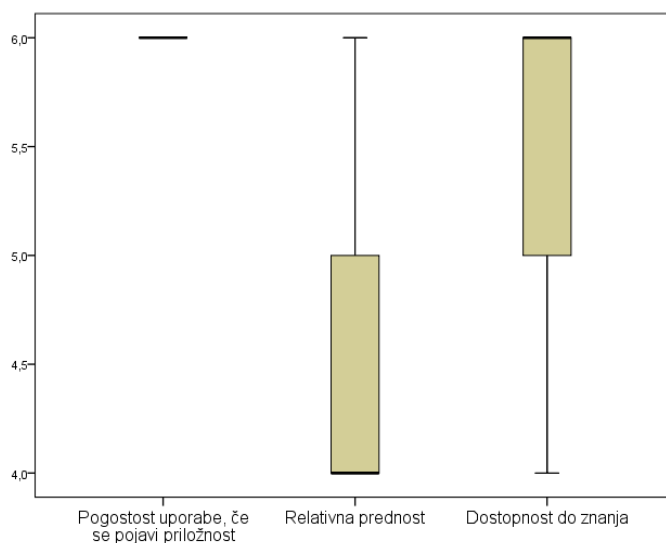
Slika 4.7: Prikaz razporeditve ocen sociološkega vidika za aktivnost Q15.

pravljanja hroščev in morebitnih novih funkcionalnosti. Testiranje se izvaja tako, da se uporabi testni načrt, nato pa se ponovno stestira vse testne scenarije in primere prejšnjih funkcionalnosti. Večinoma se testiranje izvaja ročno na mobilnih napravah. Aktivnost je ocenjena kot neuporabna, saj je tehnično neustrezna in sociološko nesprejeta. Zaradi visokega ekonomskega vpliva je aktivnost smiselno izboljšati. Razlog za tehnično neustreznost je pomanjkanje avtomatiziranega testiranja, razlog za sociološko nesprejetost pa je v zamudnosti testiranja, saj je bila najslabše ocenjena karakteristika relativna prednost. Razporeditev ocen sociološke komponente po posamezni karakteristiki je predstavljena na Sliki 4.7, na kateri lahko vidimo, da med ocenami ni prevelikega odstopanja. Za izboljšanje sociološke ocene smo predlagali, da se za regresijsko testiranje namesto celovitega testnega načrta uporabi ločen načrt za regresijsko testiranje. Regresijski načrt naj zajema le ključne testne scenarije določenih funkcionalnosti. Podobno kot pri aktivnosti (Q4) Priprava testnega načrta bi lahko za pisanje in izvajanje regresijskih testov uporabljali orodje TestRail, kar je eden iz-

med možnih ukrepov za povečanje tehnične ustreznosti. Drugi ukrep za povečanje tehnične ustreznosti je avtomatizacija regresijskih testov, podobno kot smo predlagali pri aktivnosti (Q14) Testiranje uporabniškega vmesnika. Vodstvo in tehnična vodja so sprejeli ukrep za pripravo ločenega regresijskega načrta. Zaradi negativnega ekonomskega vpliva v primeru nakupa licenc za orodje TestRail, ki smo ga omenili že pri aktivnosti (Q4) Priprava testnega načrta, bo podjetje iskalo podobno orodje, ki bo na voljo brezplačno. Zavrnjen je bil predlog za avtomatizacijo regresijskega testiranja, saj pravijo, da mobilnih aplikacij ne nadgrajujejo dovolj pogosto, da bi upravičili vire za razvoj avtomatskih regresijskih testov.

- **(Q17) Testiranje potisnih sporočil**

Potisna sporočila spodbujajo uporabnike, da ponovno odprejo mobilno aplikacijo. Kako pogosto uporabniki uporabljajo mobilno aplikacijo, je eden izmed ključnih indikatorjev uspeha mobilne aplikacije, saj se tako povečuje dobiček. Zato ima aktivnost razvoj in testiranje potisnih sporočil nadpovprečen ekonomski vpliv na podjetje. Aktivnost je bila identificirana kot neuporabna, torej sociološko nesprejeta in tehnično neustrezna. Tehnična vodja sta kot razlog za tehnično neustreznost navedla, da je koda, ki omogoča pošiljanje potisnih sporočil, zelo slaba in ni v skladu z notranjimi standardi in informacijskimi tehnologijami. Posledično je tudi testiranje tehnično neustrezno. Razlog za sociološko nesprejetost je v tem, da je testiranje zaradi slabe implementacije zelo zamudno. Na Sliki 4.8 lahko vidimo, da je bila porazdelitev ocen za karakteristiko pogostost uporabe, če se pojavi priložnost, enakomerna, medtem ko je bilo za ostali dva karakteristiki malce več odstopanja, vendar je večina uporabnikov podala podobno oceno. Kot ukrep smo tehničnima vodjema in direktorju predlagali, naj kodo za pošiljanje potisnih sporočil refaktorirajo, da bo v skladu z vsemi standardi ter se jo bo dalo lažje stestirati. S tem ukrepom bi dvignili tako oceno tehnične kot tudi sociološke komponente. Zaradi virov, ki jih zahteva



Slika 4.8: Prikaz razporeditve ocen sociološkega vidika za aktivnost Q17.

refaktorizacija, bi se znižala ocena ekonomske komponente, vendar bi obenem zaposleni pridobili dodatno znanje o dobrih praksah programiranja, kar bi zopet povečalo oceno ekonomske komponente. Kot drugi možni ukrep predlagamo uvedbo zunanjega orodja oziroma platforme, ki omogoča pošiljanje potisnih sporočil znotraj aplikacije. S tem bi povečali tudi tehnično ustreznost, saj slaba koda ne bi bila več v uporabi, takšna orodja pa po navadi omogočajo preprosto testiranje, kar bi povečalo uporabniško sprejetost. Nakup licenc za uporabo orodij bi predstavljal strošek, s čimer bi se zmanjšala ekonomska ocena. Direktor in tehnična vodja so prvi ukrep deloma zavrnili, v bližnji prihodnosti namreč nimajo na voljo dovolj virov za refaktorizacijo kode. Tudi drugi ukrep je bil zavrnjen, saj ne želijo uvesti novega orodja, ki bi omogočalo pošiljanje potisnih sporočil, ker takšne funkcionalnosti raje razvijajo sami.

4.3.2 Podjetje 2

V okviru izvedene študije primera so zaposleni v Podjetju 2 ocenili aktivnosti in orodja iz Tabele 4.2.2 z razvitim ogrođjem, ki smo ga predstavili v Poglavju 3.4. Rezultati ocenjevanja posameznih aktivnosti in orodij so prikazani v Tabeli 4.2. Člani razvojne ekipe so ocenjevali sprejetost posamezne aktivnosti in orodja, tehnični vodja je ocenjeval njihovo tehnično ustreznost, direktor pa vpliv aktivnosti in orodij na uspešnost podjetja. Rezultati vrednotenja posameznih aktivnosti in orodij se predstavljajo na Sliki 4.9. Na razsevnem diagramu lahko vidimo, da se je manj kot polovica aktivnosti in orodij uvrstila v četrti kvadrant. Ostale aktivnosti, ki se ne nahajajo v četrtem kvadrantu, imajo vsaj eno od dimenzij ocenjeno podpovprečno in jih je možno izboljšati. Z direktorjem smo se dogovorili, da bomo podali izboljšave za tiste aktivnosti in orodja, ki se ne nahajajo v četrtem kvadrantu in imajo visoko oceno z ekonomskega vidika. Direktor je povedal, da je izboljšava takšnih aktivnosti in orodij najbolj ključna za podjetje, saj imajo visok vpliv na uspešnost podjetja. Aktivnosti in orodja s podpovprečnim ekonomskim vplivom so na razsevnem diagramu na Sliki 4.9 označena z modro bravo. Aktivnosti in orodja z nadpovprečnim ekonomskim vplivom pa so označena z rdečo barvo. Točna vrednost ekonomske ocene je podana v kvadratu zraven točke.

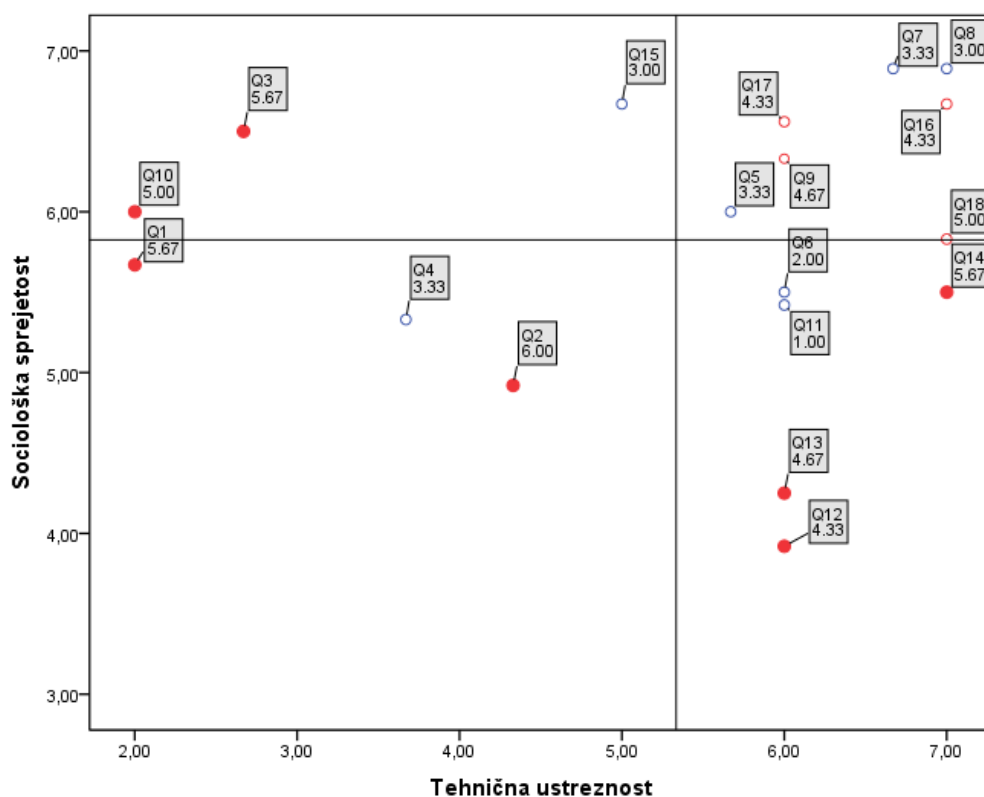
V nadaljevanju smo predstavili izbrane aktivnosti in orodja, ki smo jih v dogovoru s podjetjem poskusili izboljšati. Te aktivnosti so morale imeti visoko oceno ekonomskega vidika ter podpovprečno oceno s tehničnega in/ali z uporabniškega vidika. Aktivnosti in orodja, ki so izbrani kandidati za izboljšanje, so na razsevnem diagramu označeni s točkami z rdečim polnilom.

- **(Q1) Iskanje idej in konceptov ter (Q3) Izdelava prototipa**

Obe aktivnosti se izvajata na začetku razvojnega procesa mobilne igre. Prva aktivnost predstavlja iskanje idej in konceptov za samo mobilno igro kot tudi za posamezne funkcionalnosti. Nato se izbere najboljše

	Tehnični vidik				Sociološki vidik				Ekonomski vidik			
E	T1	T2	T3	PT	S1	S2	S3	PS	E1	E2	E3	PE
Q1	2	2	2	2	6	5.25	5.75	5.67	6	6	5	5.67
Q2	3	5	5	4.33	3.5	5	6.25	4.92	6	6	6	6
Q3	2	2	4	2.67	6.5	6.33	6.67	6.5	6	5	6	5.67
Q4	2	6	3	3.67	4.5	5.75	5.75	5.33	6	3	1	3.33
Q5	5	7	5	5.67	6	6	6	6	2	2	6	3.33
Q6	6	6	6	6	5.25	5.5	5.75	5.5	2	2	2	2
Q7	7	7	6	6.67	7	7	6.67	6.89	2	2	6	3.33
Q8	7	7	7	7	7	7	6.67	6.89	5	2	2	3
Q9	6	6	6	6	6	6.5	6.5	6.33	6	6	2	4.67
Q10	2	2	2	2	5.75	6	6.25	6	6	5	4	5
Q11	6	6	6	6	5.5	5.75	5	5.42	1	1	1	1
Q12	6	6	6	6	3.75	3.75	4.25	3.92	6	6	1	4.33
Q13	6	6	6	6	3.75	4.75	4.25	4.25	6	6	2	4.67
Q14	7	7	7	7	6.33	5	4.33	5.22	6	6	5	5.67
Q15	4	7	4	5	6.5	6.75	6.75	6.67	5	2	2	3
Q16	7	7	7	7	6.67	6.33	6.67	6.56	6	5	2	4.33
Q17	6	6	6	6	6.25	6.25	6.5	6.33	6	5	2	4.33
Q18	7	7	7	7	5.75	6	5.75	5.83	5	5	5	5
P.				5.33				5.79				4.13

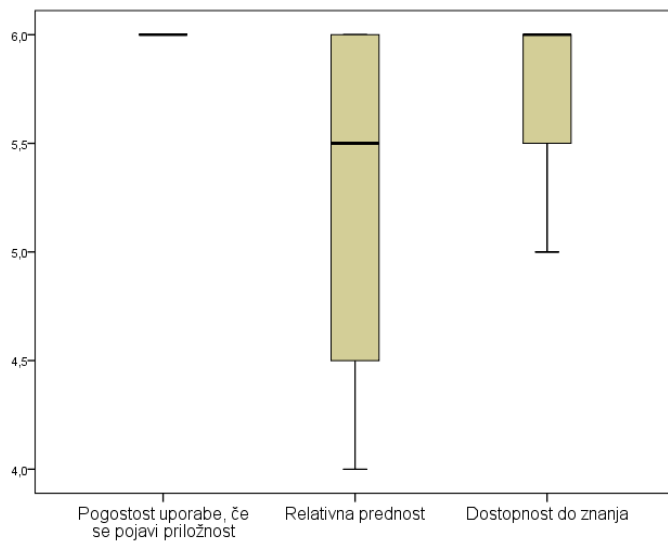
Tabela 4.2: Rezultati ocenjevanja posameznih aktivnosti in orodij v Podjetju 2. Stolpec **E** označuje aktivnosti in orodja po Tabeli 4.2.2. Stolpci **T1**, **T2** in **T3** predstavljajo povprečno vrednost karakteristik tehničnega vidika: skladnost s standardi informacijskih tehnologij, vpliv na organizacijo in skladnost s tehničnimi potrebami podjetja. Stolpci **S1**, **S2** in **S3** predstavljajo povprečno vrednost karakteristik sociološkega vidika: pogostost uporabe, če se pojavi priložnost, relativna prednost in dostopnost do znanja. Stolpci **E1**, **E2** in **E3** predstavljajo povprečno vrednost karakteristik ekonomskega vidika: čiste koristi, zadovoljstvo kupcev, strokovno učenje. Stolpci **TP**, **SP** in **EP** predstavljajo povprečno vrednost vseh karakteristik danega vidika za posamezno aktivnost ali orodje.



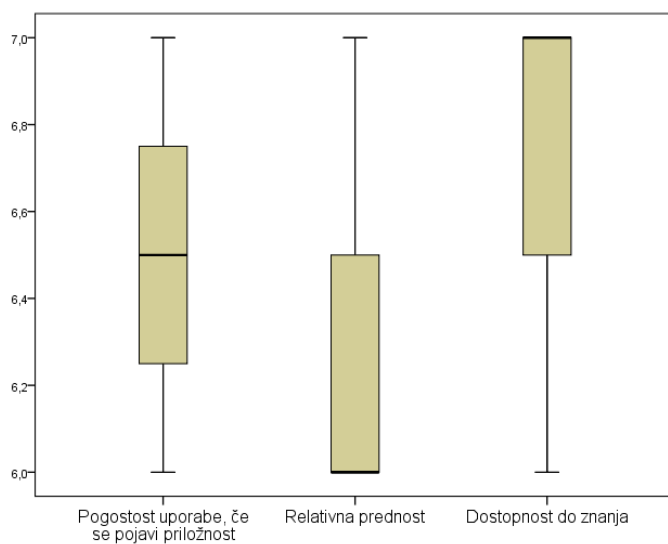
Slika 4.9: Grafični prikaz ovrednotenih aktivnosti in orodij v Podjetju 2.

ideje in na njihovi podlagi izdelava zgodnje prototipe. Ekonomski vpliv obeh aktivnosti je nadpovprečen, saj sta dobra ideja in prototip prva koraka za uspešno mobilno igro. Tehnični vodja je obe aktivnosti označil kot neustrezni, aktivnost iskanje idej in konceptov pa ima tudi malenkost podpovprečno oceno iz uporabniške sprejetosti. Razlog za dano oceno je, da se aktivnosti izvajata na staromodni način, in sicer s pisanjem in risanjem na papir in tablo. Uporabnikom se zdi pisanje na papir zamudno in zelo nepregledno. Obe aktivnosti sta zato kandidata za izboljšanje.

Razporeditev ocen posameznih karakteristik za aktivnosti je predstavljena na slikah 4.10 in 4.11. Vidimo, da je največje odstopanje pri



Slika 4.10: Prikaz razporeditve ocen sociološkega vidika za aktivnost Q1.

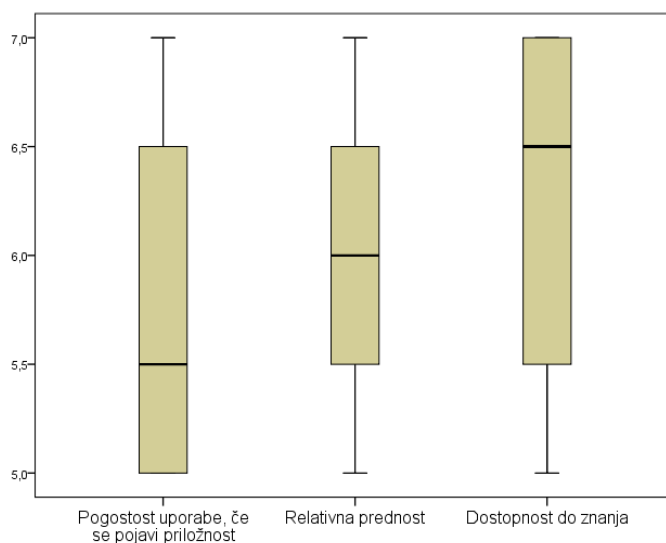


Slika 4.11: Prikaz razporeditve ocen sociološkega vidika za aktivnost Q3.

ocenah pri uporabniško nesprejeti aktivnosti Q1, in sicer pri karakteristiki relativna prednost. Pri ostalih karakteristikah so imeli uporabniki podobno mnenje in so podali podobne ocene. Kot ukrep smo predlagali uporabo tehničnega orodja, ki bi bilo skladno z informacijskimi standardi in tehničnimi potrebami podjetja. Kot primer orodja za beleženje idej smo predlagali rešitev Innovation Cloud [78], s katerim uporabniki podajajo ideje in medsebojno sodelujejo pri izbiri najboljše ideje. Kot primer orodja za izdelavo prototipov in diagramov poteka smo predlagali orodje [79], ki pohitri izdelavo prototipov in omogoča večjo preglednost. Predlagali smo brezplačno različico obeh orodij za ohranitev nadpovprečnega ekonomskega vpliva. Oba predloga je vodstvo pogojno sprejelo – to pomeni, da bodo brezplačno različico orodja poskusno uporabljali en mesec in nato ocenili njegov prispevek.

- **(Q10) Testiranje uporabniškega vmesnika s simulatorjem Unity**

Aktivnost testiranje uporabniškega vmesnika v podjetju pomeni preverjanje ustreznosti pisave, postavitve grafičnih elementov in preverjanje lokalizacij na različnih scenah in dialogih mobilne igre. Testiranje poteka s simulatorjem Unity, pri čemer uporabnik v simulatorju nastavi več različnih razmerij stranic zaslona in izvede testiranje. Aktivnost je bila ocenjena kot tehnično neustrezna, vendar je med razvojno ekipo dobro sprejeta. Ker ima visok ekonomski vpliv, je dober kandidat za izboljšanje. Razporeditev ocen uporabniške sprejetosti je predstavljena na Sliki 4.12, na kateri vidimo, da se ocene posamezne karakteristike gibljejo v enakem razponu, in sicer med pet in sedem. Uporabniki menijo, da imajo dovolj znanja za izvajanje aktivnosti. Morebiten razlog za nadpovprečno ocenjeno karakteristiko relativna prednost je v tem, da so morali uporabniki pred uvedbo testiranja s simulatorjem ročno testirati na številnih mobilnih napravah, kar je zelo zamudno. Težava tehnične neustreznosti pa je v tem, da je testiranje nezanesljivo, saj zgolj nastavitve različnih razmerij stranic ne pokrije vseh primerov, ker se je treba osredotočiti tudi na ločljivost. Nastavitve ločljivosti

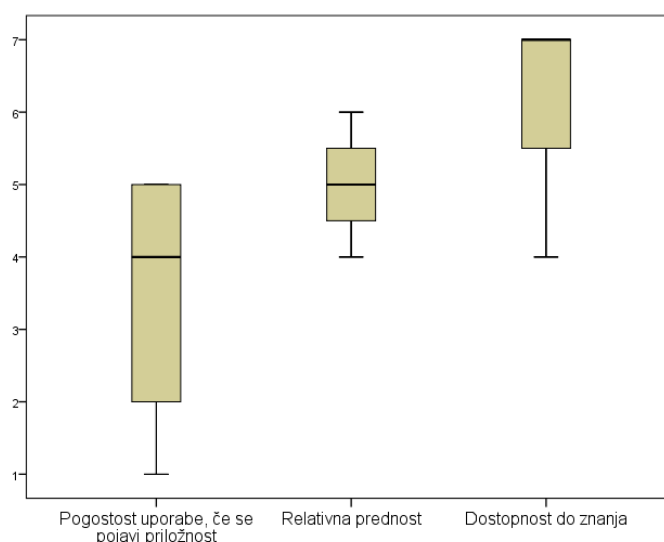


Slika 4.12: Prikaz razporeditve ocen sociološkega vidika za aktivnost Q10.

podpira tudi program Unity, vendar se uporaba teh nastavitev uporabnikom zdi zamudna in bi s takšnim ukrepom znižali sprejetost. Zato predlagamo uvedbo avtomatskega testiranja uporabniškega vmesnika z orodjem Appium[80], ki omogoča avtomatsko testiranje na platformah Android in iOS, pri čemer je možno avtomatsko testiranje na dejanskih mobilnih napravah. Z uvedbo novega orodja bi bilo testiranje zanesljivejše in bi odpravilo več napak, s čimer bi močno povečali tehnično ustreznost. Manjša pomanjkljivost je, da bi uvedba na začetku negativno vplivala na ekonomski vidik, saj bi razvoj avtomatskih testov zahteval določene vire. Sčasoma bi se ekonomski vpliv zopet povečal, saj bi bilo testiranje avtomatizirano, kar bi občutno zmanjšalo čas testiranja. S tem bi tudi ohranili ali celo povečali uporabniško sprejetost, saj ne bi bila več tako zamudna. Predlog so vodstvo in tehnični vodja sprejeli, vendar bodo z razvojem avtomatskih testov začeli šele pri razvoju nove mobilne igre. Razlog za to je, da trenutno ne želijo spreminjati kode obstoječih projektov.

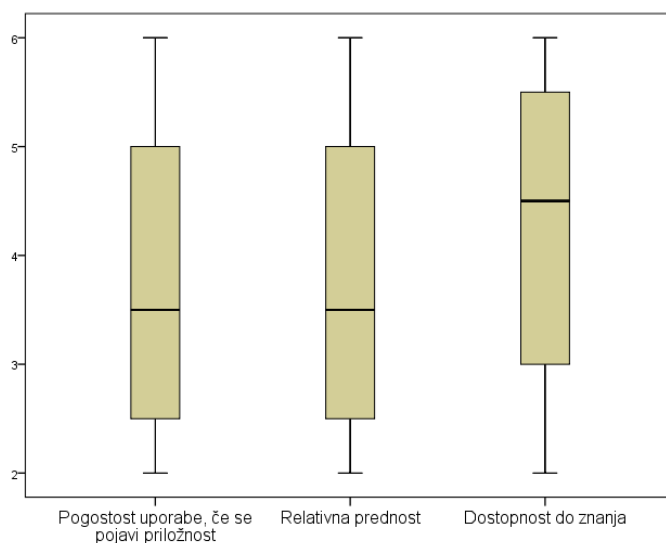
- **(Q2) Priprava tehnične specifikacije**

Aktivnost priprava tehnične specifikacije v primeru Podjetja 2 pomeni zapis razvoja posamezne funkcionalnosti v dokument v oblaku. Pri pisanju tehnične specifikacije naj bi sodelovala celotna ekipa in s tem prispevala k izboljšanju kakovosti. Aktivnost ima velik ekonomski vpliv, zato jo je smiselno izboljšati. Aktivnost je z vidika tehnične ustreznosti podpovprečno ocenjena, predvsem zaradi neskladnosti specifikacij. Dokumentacija je dostikrat pomanjkljiva, kar se opazi šele v procesu razvoja, zato jo je treba spremeniti tekom razvoja. To pripelje do nesprejetosti razvojne ekipe, saj jim slaba specifikacija povzroča dvojno delo.



Slika 4.13: Prikaz razporeditve ocen sociološkega vidika za aktivnost Q2.

Iz Slike 4.13 je razvidno, da nekateri uporabniki aktivnost izvajajo zelo redko ali pa sploh ne. Kot ukrep smo predlagali, da se uvede enotna predloga za pisanje specifikacije, ki bo zagotovila, da so pokriti vsi vidiki razvoja funkcionalnosti. S tem bi povečali njeno tehnično ustreznost. Da bi povečali uporabniško sprejetost, smo predlagali, da se pred začetkom razvoja funkcionalnosti skliče sestanek za udeležence

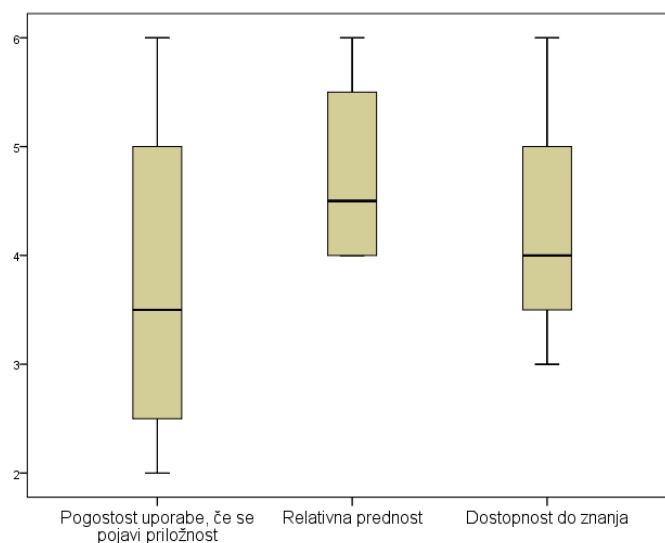


Slika 4.14: Prikaz razporeditve ocen sociološkega vidika za aktivnost Q12.

pri razvoju. Na sestanku bi strokovno pregledali specifikacijo, odpravili morebitne pomanjkljivosti in razjasnili nejasnosti ter še dodatno poudarili pomembnost pisanja dokumentacije. Sprejet je bil le prvi ukrep, medtem ko je tehnični vodja drugega zavrnil. Razlog za zavrnitev je, da si ne želijo dodatnih sestankov, saj se to tehničnemu vodju zdi zamudno.

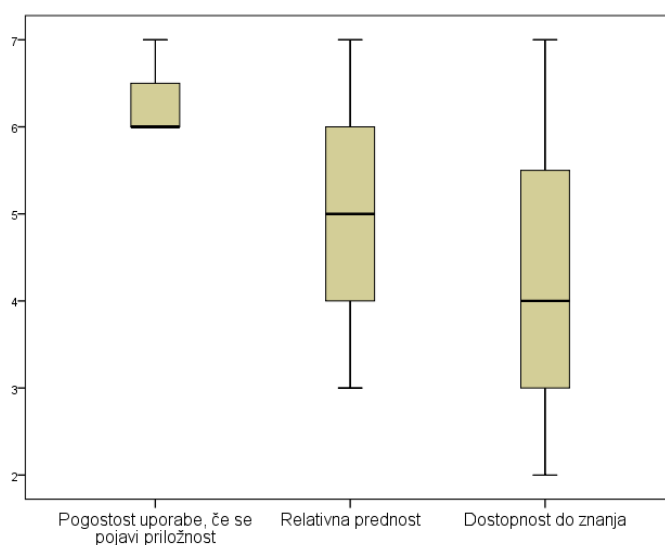
- **(Q12) Uporaba orodja Crashlytics za iskanje napak in (Q13) Uporaba orodja Swrve za testiranje**

Uporaba obeh orodij je bila ocenjena podobno, in sicer imata orodji visok ekonomski vpliv, sta tehnično ustrezni, vendar s strani uporabnikov slabo sprejeti. Orodje Crashlytics se uporablja za iskanje napak na produkcijskih različicah mobilne aplikacije in zabeleži nenadne zaustavitve (ang. *crash*) aplikacije. Z orodjem Swrve pa izvajajo A/B testiranje, pri čemer uporabnikom predstavijo več različic funkcionalnosti, da določijo najboljšo, poleg tega pa služi za testiranje in analizo po-



Slika 4.15: Prikaz razporeditve ocen sociološkega vidika za aktivnost Q13.

datkov. Z vrednotenjem posameznih karakteristik, ki so predstavljene na slikah 4.14 in 4.15, smo ugotovili, da pogostost uporabe orodij, če se pojavi priložnost, zelo niha. Orodji nekateri uporabniki zelo redko uporabljajo, saj se raje poslužujejo uporabe drugih orodij, ki so jim bolj poznana in rešujejo bolj specifične težave. Poleg tega smo ugotovili, da se zdijo orodja nekaterim uporabnikom nejasna, saj nimajo dovolj znanja za njihovo uporabo. Karakteristika relativna prednost je dobila malenkost boljšo oceno pri aktivnosti Q13 kot pri Q12. Vodstvu in tehničnemu vodju smo predlagali, naj organizirajo delavnico, na kateri bodo predstavili prednosti uporabe obeh orodij in razložili delovanje orodij, da bo ekipa pridobila potrebna znanja za njihovo uporabo. Sklep je bil sprejet in podjetje bo v prihodnosti organiziralo delavnico. Vendar tehnični vodja pravi, da razvojne ekipe ne bodo silili v uporabo teh dveh orodij, če bodo sami menili, da lahko z drugimi orodji, ki so jim ljubša, dosežejo enak rezultat.



Slika 4.16: Prikaz razporeditve ocen sociološkega vidika za aktivnost Q14.

- **(Q14) Testiranje uporabniške izkušnje**

Testiranje uporabniške izkušnje se v podjetju izvaja po vsakem večjem mejniku razvoja mobilne igre. Uporabniško izkušnjo testirajo v obliki fokusnih skupin, znotraj katerih izvajajo intervjuje, da bi pridobili čim več koristnih informacij o tem, kako izdelek vidijo uporabniki. Aktivnost se nahaja v tretjem kvadrantu, kar pomeni, da je tehnično primerna, saj se intervjuji s fokusnimi skupinami izvajajo v tehnično dovršeni sobi, vendar je sociološko nesprejeta. Razlog za podpovprečno oceno sociološkega vidika je v tem, da je aktivnost zamudna, ker intervjuji trajajo tudi po več ur, saj se diskusije oddaljijo od prvotne teme. Poleg zamudnosti pa, kot je razvidno iz razporeditve ocen na Sliki 4.16, nekateri uporabniki aktivnosti menijo, da za izvajanje aktivnosti nimajo dovolj znanja. Karakteristika pogostost uporabe, če se pojavi priložnost, so vsi uporabniki dali visoko oceno. Ker ima visoko oceno ekonomskega vpliva, je kandidat za izboljšanje. Kot ukrep za povečanje sociološke ocene smo vodstvu predlagali, naj dodatno izobrazijo kader o tem, kako

testirati uporabniško izkušnjo in kaj je njen namen. Poleg tega naj bodo intervjuji s fokusnimi skupinami bolj strukturirani in moderirani, da se izogne večurnim intervjujem. Izobraževanje bi predstavljalo strošek, s čimer bi se zmanjšale čiste koristi, vendar bi zaposleni pridobili nova znanja, zaradi česar bi ekonomska ocena ostala nespremenjena. Predlog je vodstvo sprejelo.

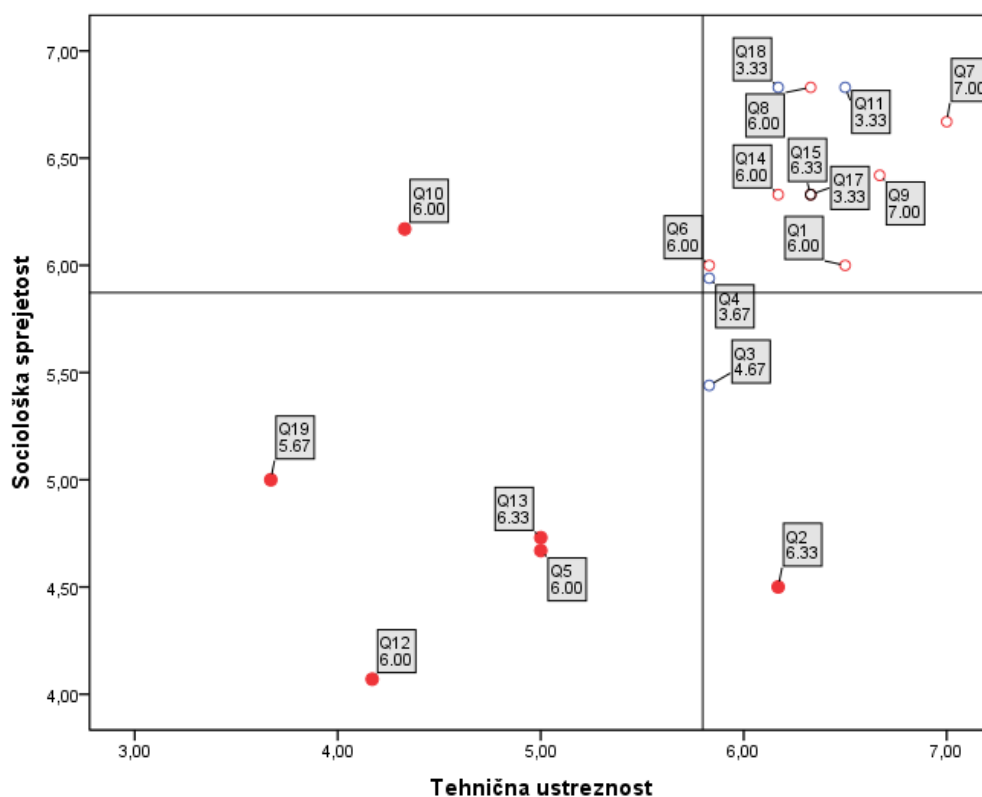
4.3.3 Podjetje 3

Aktivnosti in orodja iz Tabele 4.2.3 so zaposleni v Podjetju 3 ovrednotili na podlagi ogrodja, ki smo ga predstavili v Poglavlju 3. Rezultati ocenjevanja posameznih aktivnosti in orodij so prikazani v Tabeli 4.3. S pomočjo vprašalnikov so člani razvojnih ekip ocenili sprejetost metodologije, tehnična vodja njeno tehnično ustreznost, direktor pa vpliv na uspešnost podjetja. Rezultati ocenjevanja aktivnosti in orodij so predstavljeni na Sliki 4.17. S preučevanjem razsevnega diagrama smo ugotovili, da se je kar veliko število aktivnosti in orodij uvrstilo v četrti kvadrant. To pomeni, da so jih uporabniki sprejeli in so tehnično ustrezna. Druge aktivnosti, ki se ne nahajajo v četrtem kvadrantu, imajo vsaj eno izmed dimenzij ocenjeno podpovprečno in jih je treba izboljšati. Po pogovoru z vodstvom podjetja smo se odločili, da je za podjetje najpomembnejše, da se naslovijo aktivnosti in orodja, ki imajo visoko oceno z ekonomskega vidika. Poleg visoke ocene ekonomskega vidika, ki pomeni velik vpliv na uspešnost podjetja, mora imeti aktivnost ali orodje podpovprečno oceno s tehničnega in/ali sociološkega vidika. Aktivnosti in orodja s podpovprečnim ekonomskim vplivom so na razsevnem diagramu označena z modro barvo, medtem ko so tista z nadpovprečnim ekonomskim vplivom označena z rdečo barvo.

V nadaljevanju smo predstavili izbrane aktivnosti in orodja, ki so bila vsaj z enega vidika ocenjena podpovprečno in imajo visok ekonomski vpliv. Za lažjo predstavitev aktivnosti in orodij, ki jih bomo poskušali izboljšati, imajo njihove točke rdeče polnilo in so večje v primerjavi z ostalimi točkami na razsevnem diagramu.

	Tehnični vidik				Sociološki vidik				Ekonomski vidik			
E	T1	T2	T3	PT	S1	S2	S3	PS	E1	E2	E3	PE
Q1	6	6.5	7	6.5	5.83	5.83	6.33	6	7	7	4	6
Q2	6	6	6.5	6.17	3.17	4.67	5.67	4.5	7	5	7	6.33
Q3	5.5	6.5	5.5	5.83	4.67	5.67	6	5.44	4	6	4	4.67
Q4	6	5.5	6	5.83	6.17	5.67	6	5.94	4	4	3	3.67
Q5	5	5.5	4.5	5	3.67	5	5.33	4.67	7	7	4	6
Q6	5.5	6	6	5.83	6	5.8	6.2	6	7	7	4	6
Q7	7	7	7	7	7	7	6	6.67	7	7	7	7
Q8	6	6.5	6.5	6.33	7	7	6.5	6.83	6	7	5	6
Q9	7	6	7	6.67	6.5	6.75	6	6.42	7	7	7	7
Q10	4	4.5	4.5	4.33	5.75	6.5	6.25	6.17	6	7	5	6
Q11	6.5	6.5	6.5	6.5	7	6.75	6.75	6.83	3	2	5	3.33
Q12	4	4.5	4	4.17	4.2	4.2	3.8	4.07	6	6	6	6
Q13	5	5	5	5	5.8	5.4	3	4.73	7	7	5	6.33
Q14	6	6.5	6	6.17	6.33	6	6.67	6.33	7	6	5	6
Q15	6.5	6	6.5	6.33	6.5	5.75	6.75	6.33	7	7	5	6.33
Q16	6.5	7	5.5	6.33	6.75	6.25	6.5	6.5	6	7	5	6
Q17	6.5	5.5	7	6.33	6.67	5.67	6.67	6.33	4	4	2	3.33
Q18	6	6.5	6	6.17	7	6.5	7	6.83	4	4	2	3.33
Q19	4	4	3	3.67	5	7	5	5	5	6	6	5.67
P.				5.80				5.87				5.53

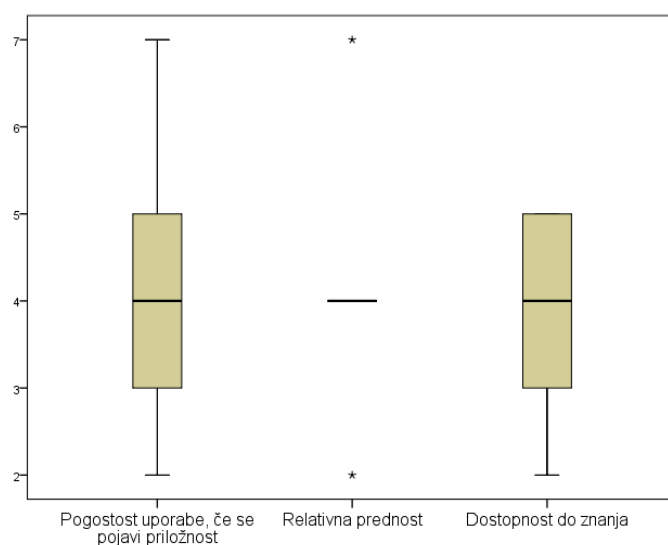
Tabela 4.3: Rezultati ocenjevanja posameznih aktivnosti in orodij v Podjetju 3. Stolpec **E** označuje aktivnosti in orodja po Tabeli 4.2.3. Stolpci **T1**, **T2** in **T3** predstavljajo povprečno vrednost karakteristik tehničnega vidika: skladnost s standardi informacijskih tehnologij, vpliv na organizacijo in skladnost s tehničnimi potrebami podjetja. Stolpci **S1**, **S2** in **S3** predstavljajo povprečno vrednost karakteristik sociološkega vidika: pogostost uporabe, če se pojavi priložnost, relativna prednost in dostopnost do znanja. Stolpci **E1**, **E2** in **E3** predstavljajo povprečno vrednost karakteristik ekonomskega vidika: čiste koristi, zadovoljstvo kupcev, strokovno učenje. Stolpci **TP**, **SP** in **EP** predstavljajo povprečno vrednost vseh karakteristik danega vidika za posamezno aktivnost ali orodje.



Slika 4.17: Grafični prikaz ovrednotenih aktivnosti in orodij v Podjetju 3.

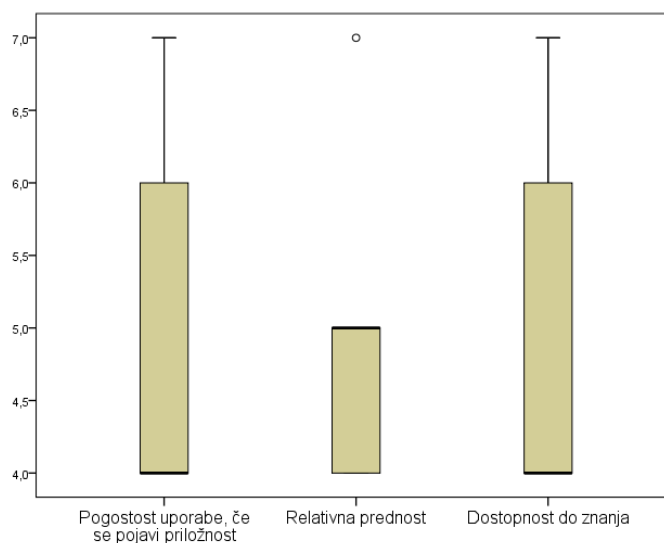
- (Q12) Uporaba orodja Jenkins za pregledovanje kode

Orodje Jenkins in njegovi vtičniki se uporabljajo za pregledovanje kode. Razvijalci si med seboj pregledujejo kodo in tako odkrijejo napake in pomanjkljivosti v kodi. Orodje ima visok ekonomski vpliv, vendar zelo nizko sprejetost med uporabniki, nizka pa je tudi tehnična ustreznost. Ugotovili smo, da je razlog za sociološko neustreznost v tem, da razvijalcem pregledovanje kode vzame veliko časa, oni pa bi ta čas kodo raje pisali, kot pregledovali. Razvijalci so se pritožili, da je orodje zelo toga in nima vseh funkcionalnosti, ki bi si jih želeli. Do enakega zaključka sta prišla tudi tehnična vodja, ki sta orodje ocenila kot tehnično neu-



Slika 4.18: Prikaz razporeditve ocen sociološkega vidika za aktivnost Q12.

strežno. Iz Slike 4.18 razberemo, da ocena pogostosti uporabe orodja, če se pojavi priložnost, zelo variira. Večina razvijalcev je enotnejšega mnenja glede ocenjevanja relativne prednosti, ki so jo podpovprečno ocenili vsi razvijalci, razen enega. Podpovprečno je bila ocenjena tudi karakteristika dostopnost do znanja, pri čemer je med razvijalci malce odstopanja v mnenju, koliko dostopnih virov imajo na voljo za pridobivanje znanja. V namen izboljšanja procesa pregledovanja kode smo predlagali, da se orodje zamenja s takšnim, ki bo bolj sprejeto in skladnejše s standardi in tehničnimi zahtevami. Pri uvedbi novega orodja bi bilo seveda treba izobraziti razvijalce o uporabi novega orodja. Poleg tega za povečanje sprejetosti predlagamo, da se skuša razvojno ekipo motivirati, z majhno nagrado za tistega, ki bo v enem mesecu kakovostno opravil največ pregledov kode. Z omenjenimi ukrepi bi zvišali obe dimenziji in uporaba novega orodja bi se prestavila v četrti kvadrant. Zaradi uvedbe novega orodja in bonusov bi prišlo do dodatnih stroškov, kar bi znižalo čiste koristi. Negativen vpliv na ekonomsko dimenzijo bi na dolgi rok izničili ali celo pozitivno povečali, saj bi zaposleni pri-



Slika 4.19: Prikaz razporeditve ocen sociološkega vidika za aktivnost Q19.

dobili nova znanja, boljša kakovost kode pa bi pomenila zadovoljnejše naročnike. Direktor je zavrnil predlog o bonusih, skupaj s tehničnima vodjema pa so potrdili, da bodo iskali novo orodje za pregledovanje kode ter v primeru uvedbe novega orodja ustrezno izobrazili razvijalce.

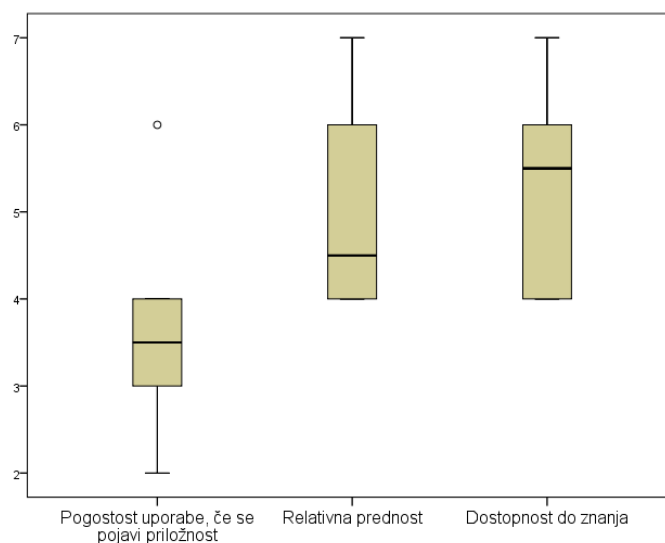
• (Q19) Notranji tedenski sestanki

Namen notranjih tedenskih sestankov je pregled poteka razvoja na projektih. Aktivnost ima visok ekonomski vpliv, zato je eden izmed kandidatov za izboljšanje. Prav tako ima zelo slabo oceno z sociološkega in tehničnega vidika, kar pomeni, da je aktivnost neustrezna in nesprejeta. Razlog za tehnično neustreznost je v tem, da sestanki potekajo v sobi, ki ne nudi nobene tehnologije, ki bi olajšala prikaz poteka razvoja na projektu, saj celoten sestanek poteka ustno. Razvojne ekipe menijo, da so sestanki v tako velikem obsegu zamudni in se prepogosto osredotočajo na podrobnosti. Na Sliki 4.19 lahko vidimo, da je večina članov razvojne ekipe podobnega mnenja, saj so podali nizko oceno

za karakteristiko relativna prednost, z izjemo enega člana, ki je podal najvišjo oceno. Ostali dve karakteristiki sta prejeli podobne ocene, ki so bile podpovprečne, vendar je bilo med ocenami malo več odstopanja. Direktorju in tehničnima vodjema smo predlagali naslednji seznam ukrepov. Za povečanje sprejetosti, naj se sestanke časovno omeji in vnaprej pripravi točno agendo, o čem se bo na sestanku razpravljalo. Pri tem je treba sestanek moderirati in takoj opozoriti, če se razpravlja o podrobnostih, kot so na primer težave, na katere je naletela razvojna ekipa. Za takšne razprave je primernejši poseben sestanek ali pa boljša komunikacija razvijalcev prek programske opreme za komuniciranje. Za povečanje tehnične ustreznosti smo predlagali nakup televizorja, na katerem bi prikazali stanje poteka obstoječih projektov glede na opravljene in neopravljene naloge, kot jih prikaže orodje Jira. Z nakupom bi malce znižali ekonomsko komponento, saj nakup televizorja predstavlja strošek, vendar bi se naložba na dolgi rok obrestovala. Z upoštevanjem ukrepov bi aktivnost prišla v četrti kvadrant in postala uporabna. Direktor in tehnična vodja se z vsemi predlogi strinjajo, sestanki so sedaj časovno omejeni, bolj osredotočeni in moderirani, za lažji prikaz stanja projektov pa se uporablja malo večji zaslon, ki prej ni bil v uporabi. To pomeni, da nakup novega televizorja ni bil potreben in ekonomska ocena je ostala nespremenjena.

- **(Q5) Priprava testnega načrta in (Q13) Testiranje funkcionalnosti mobilne aplikacije**

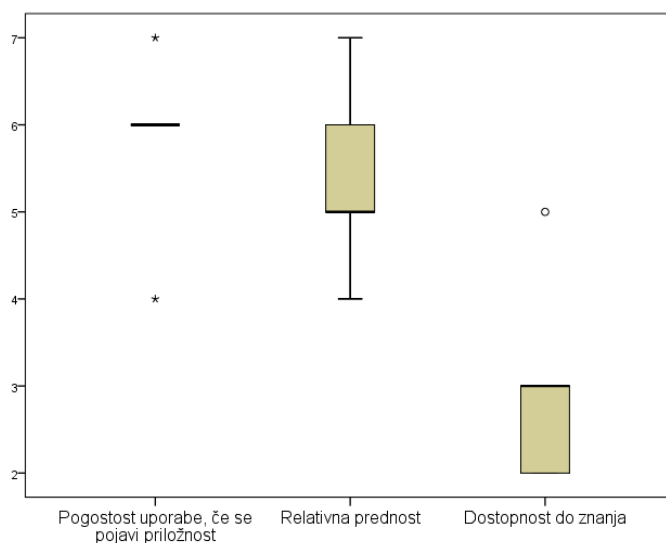
Obe aktivnosti sta podobno ocenjeni in se nahajata v prvem kvadrantu, kar pomeni, da sta nesprejeti in neustrezni. Aktivnosti sta med seboj povezani, saj težko dobro izvedemo testiranje funkcionalnosti mobilne aplikacije brez dobrega testnega načrta. Rezultat trenutne izvedbe priprave testnega načrta je dokument, v katerem so v grobem zapisane glavne funkcionalnosti, ki jih bo imela aplikacija in jih je treba testirati. Testiranje funkcionalnosti mobilne aplikacije trenutno izvajajo



Slika 4.20: Prikaz razporeditve ocen sociološkega vidika za aktivnost Q5.

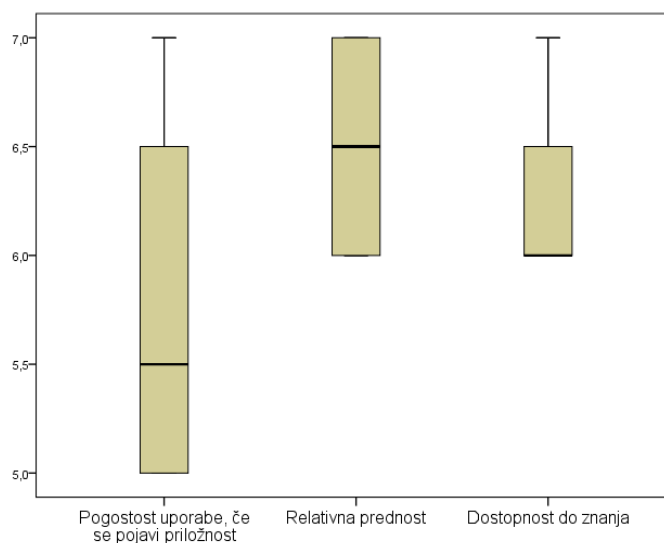
razvijalci tako, da med razvojem aplikacijo preverjajo na napravah. Aktivnosti ne moremo zavreči, saj je ena od ključnih aktivnosti, ki zagotavlja kakovost mobilnih aplikacij, kar ima visok učinek na ekonomsko komponento. Zato je nujno, da obe aktivnosti močno izboljšamo, da bosta prišli v četrti kvadrant. Na slikah 4.20 in 4.21 so predstavljene razporeditve ocen posameznih karakteristik sociološkega vidika.

Razvijalci pravijo, da za aktivnosti ne najdejo časa, poleg tega pa bi raje razvijali kot pripravljali testni načrt, kar je razvidno tudi iz zelo slabo ocenjene karakteristike pogostost uporabe za aktivnost Q5, če se pojavi priložnost. Ostali dve karakteristiki za aktivnost Q5 sta ocenjeni malce bolje, vendar še vedno podpovprečno. Menijo, da priprava testnega načrta in podrobnejše testiranje funkcionalnosti mobilne aplikacije ni njihova naloga. Pri aktivnosti Q13 je najslabše ocenjena karakteristika dostopnost do znanja, ki je z izjemo enega osamelca, prejela zelo nizke ocene. Izvedba obeh aktivnosti je bila ocenjena kot tehnično nedovršena v vseh pogledih. Kot ukrep smo predlagali, da se zaposli inženirja kakovosti, ki bo skrbel za izvedbo obeh aktivnosti. S



Slika 4.21: Prikaz razporeditve ocen sociološkega vidika za aktivnost Q13.

tem bi povečali sprejetost, saj bi razbremenili razvijalce, ki ne želijo opravljati aktivnosti. Za aktivnost testiranje funkcionalnosti mobilnih aplikacij so razvijalci ocenili, da nimajo potrebnega znanja, kar bi se rešilo z zaposlitvijo strokovnjaka. Zaposlitev novega kadra bi sprva negativno vplivala na ekonomsko dimenzijo, vendar smo prepričani, da bi se na dolgi rok zaposlitev obrestovala, saj bi povečala kakovost izdelka. Za povečanje tehnične ustreznosti predlagamo pisanje podrobnejših testnih scenarijev in testnih primerov, ki bodo pokrivali funkcionalnost aplikacije. Pisanje testnih primerov in scenarijev v dokument je zelo nepregledno, zato predlagamo, da se namesto dokumenta uporabijo razpredelnice ali pa se dodatno investira v orodje TestRail [81]. Orodje poveča učinkovitost pri pisanju testnih primerov in kasneje pri testiranju funkcionalnosti mobilne aplikacije. Direktor in tehnična vodja so sprejeli ukrep za zaposlitev inženirja kakovosti. Pravijo, da so o tem že razmišljali, po naših rezultatih pa so spoznali, da je inženir kakovosti ključen za izvajanje določenih aktivnosti. Ko bodo zaposlili inženirja kakovosti, bodo poskrbeli za podrobnejše pisanje testnih primerov in



Slika 4.22: Prikaz razporeditve ocen sociološkega vidika za aktivnost Q10.

scenarijev ter doslednejše testiranje funkcionalnosti mobilnih aplikacij. Odločili so se, da bodo za začetek uporabili razpredelnice, saj bi uvedba novega orodja poleg novega zaposlenega predstavljala prevelik strošek. Pravijo, da bodo čez nekaj mesecev skupaj z novim zaposlenim ponovno ocenili uporabo razpredelnic za pisanje testnih primerov in se morda naknadno odločili za uvedbo novega orodja.

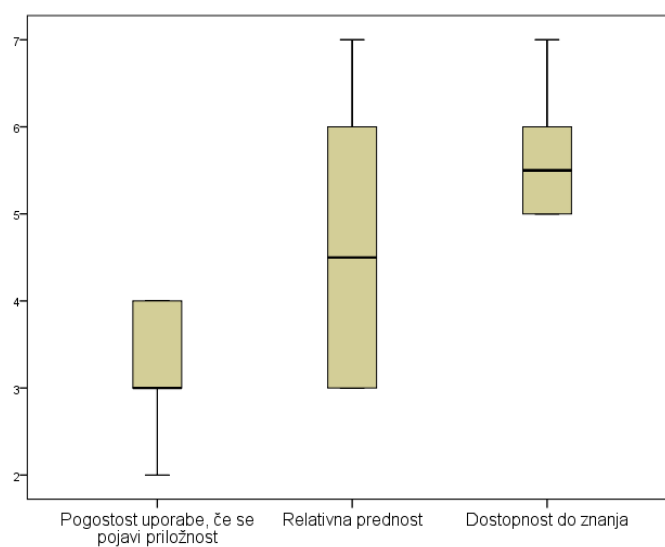
- **(Q10) Testiranje mobilne aplikacije z emulatorjem Android**

Razvijalci med delom na mobilni aplikaciji pogosto preverjajo delovanje svoje kode z osnovnim emulatorjem Android. Aktivnost so razvijalci ocenili kot nadpovprečno, medtem ko sta jo tehnična vodja ocenila kot podpovprečno. Ker ima aktivnost visok ekonomski vpliv, je kandidat za izboljšanje. Za prehod aktivnosti v četrti kvadrant je treba zvišati njeno tehnično komponento. Razporeditev ocen sociološkega vidika je predstavljena na Sliki 4.22, iz katere je razvidno, da imajo uporabniki za karakteristiki relativna prednost in dostopnost do znanja podobna

mnenja, saj sta obe prejeli podobno visoke ocene. Malce več odstopanj pri mnenju uporabnikov je pri pogostosti izvajanja aktivnosti, če se pojavi priložnost za izvajanje. Tehnična vodja sta dejala, da razvijalci porabijo veliko časa s testiranjem na osnovnem emulatorju Android, saj je emulator zelo počasen. Predlagali smo uvedbo novega, hitrejšega emulatorja Genymotion [82]. Razvijalci ga lahko uporabijo kot vtičnik v programu Android Studio, ki ga uporabljajo za razvoj mobilnih aplikacij. Uvedba novega emulatorja ne bi imela večjih posledic na sociološko komponento, močno pa bi povečala oceno tehnične ustreznosti. Direktor in tehnična vodja so predlog delno sprejeli. Za začetek se bodo odločili le za nakup ene licence in kasneje ponovno ocenili vrednosti posameznih komponent.

- **(Q2) Priprava tehnične dokumentacije**

Podjetje za pripravo tehnične dokumentacije uporablja notranjo Wikipedijo, ki vsebuje tehnično dokumentacijo. Dokumentacija obsega zapis tehnične implementacije posamezne funkcionalnosti mobilne aplikacije. Za vsak projekt in vsako novo funkcionalnost se uporabi enaka predloga za pisanje. Zato je bila aktivnost ocenjena kot tehnično ustrezna, vendar so jo uporabniki slabo sprejeli. Razvojni ekipi se zdi pisanje tehnične dokumentacije zamudno, saj se zahteve dostikrat spreminjajo že po začetku razvoja, zaradi česar je pogosto treba posodabljati tehnično dokumentacijo. To pripelje do slabe pogostosti uporabe, če se pojavi priložnost, kar je razvidno iz Slike 4.23, kjer so uporabniki podali podobno nizko oceno za omenjeno karakteristiko. V nasprotju je karakteristika dostopnost do znanja prejela zelo visoko oceno. Malce več odstopanja pri mnenju uporabnikov je bilo pri ocenjevanju relativne prednosti, saj je prejela visok razpon ocen. Ker se zahteve spreminjajo zaradi naročnikovih želja, česar ni mogoče preprečiti, smo kot ukrep predlagali sklic sestanka celotne ekipe, na katerem bi predstavili pomembnost in prednosti dobre tehnične dokumentacije ter s tem skušali motivirati razvojno ekipo. Poleg tega smo za dodatno motivacijo pre-



Slika 4.23: Prikaz razporeditve ocen sociološkega vidika za aktivnost Q2.

dlagali nagrado za razvijalca, ki bo največ prispeval k pisanju dokumentacije. Prvi predlog so vodstvo in tehnična vodja sprejeli, drugega pa zavrnili.

Poglavje 5

Sklepne ugotovitve

V magistrskem delu smo izdelali ogrodje za vrednotenje aktivnosti in orodij za testiranje, ki se uporabljajo v podjetjih, ki se ukvarjajo z razvojem mobilnih aplikacij. V Poglavju 2 smo pregledali literaturo in ključna področja. Ključna področja so obstoječa ogrodja za vrednotenje procesov razvoja programske opreme in testiranja programske opreme ter značilnosti in dobre prakse pri testiranju mobilnih aplikacij. V Poglavju 3 smo zgradili naše ogrodje in ga z izvedbo študije primera v treh podjetjih preverili v praksi, kar je predstavljeno v Poglavju 4. Z omenjenim smo dosegli vse cilje, ki smo si jih zastavili v uvodu magistrskega dela.

Prva dva zadana cilja, pregled obstoječih ogrodij za vrednotenje procesa razvoja programske opreme in izboljšavo testiranja ter pregled obstoječih dobrih praks pri testiranju mobilnih aplikacij smo uresničili v Poglavju 2. Navedli in opisali smo nekatera ogrodja za vrednotenje procesa razvoja programske opreme, ki vključujejo tehnični, sociološki in ekonomski vidik ter zmožnostne in zrelostne modele testiranja. Predstavili smo tudi obstoječe dobre prakse pri testiranju programske opreme in mobilnih aplikacij ter vrsto orodij, ki se uporabljajo pri testiranju.

Tretji cilj, razvoj celovitega ogrodja za vrednotenje aktivnosti in orodij pri testiranju mobilnih aplikacij, smo uresničili v Poglavju 3. Model, ki omogoča celovito vrednotenje procesa testiranja mobilnih aplikacij s tehničnega, soci-

ološkega in z ekonomskega vidika, smo zgradili na podlagi relevantne literature iz Poglavja 2.

Četrty in peti cilj smo uresničili v Poglavju 4. Naše teoretično ogrodje smo preizkusili v praksi s študijami primera v treh slovenskih podjetjih, ki se ukvarjajo z razvojem mobilnih aplikacij ali iger. Z njegovo pomočjo smo v podjetjih identificirali tehnično neustrezne, sociološko nesprejete ter ekonomsko vplivne aktivnosti in orodja. Pri tem je bil vsak vidik sestavljen iz treh karakteristik, kar zagotavlja celovitost posameznega vidika. Dobljene podatke smo analizirali in vodstvu podjetij podali koristne informacije za izboljšavo posameznih aktivnosti in orodij.

Ogrodje se je izkazalo kot uspešno, saj vsa tri podjetja, v katerih smo opravljali študijo primera, menijo, da ogrodje nudi koristne informacije o izboljšavi procesa razvoja in testiranja mobilnih aplikacij.

Čeprav je model uspešno preverjen v treh podjetjih, kar dokazuje njegovo splošno učinkovitost, ima določene omejitve. Preverjeno je bilo le v manjših podjetjih, zato bi ga bilo v prihodnosti smiselno preveriti tudi v večjih slovenskih podjetjih in v podjetjih zunaj Slovenije.

Literatura

- [1] D. Avison, G. Fitzgerald, “Information System Development: Methodologies, Techniques and Tools”, 4th Edition, McGraw-Hill, 2006.
- [2] D. Vavpotič, T. Hovelja, “Improving the Evaluation of Software Development Methodology Adoption and its Impact on Enterprise Performance”, v zborniku: Computer Science and Information Systems, št. 8, zv. 1, str. 165–187, 2012.
- [3] W. Afzal, S. Alone, K. Glocksien, R. Torkar, “Software test process improvement approaches: A systematic literature review and an industrial case study”, v zborniku: Journal of Systems and Software, št. 111, str. 1–33, 2016.
- [4] IBM, “Rational Unified Process v7.0.1 (IBM Rational Method Composer plugin)”, Elektronski priročnik, 2006.
- [5] Statista, “Number of apps available in leading app stores as of July 2015”. Dostopno na: <http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>
- [6] App Annie, “App Annie Index: Market Q1 2016”. Dostopno na: <http://blog.appannie.com/app-annie-index-market-q1-2016/>
- [7] D. Knott, “Hands-On Mobile App Testing: A guide for Mobile Testers and Anyone Involved in the Mobile App Business”, 4th Edition, Addison-Wesley, 2015.

-
- [8] H. Muccini, A. Di Francesco, P. Esposito, “Software testing of mobile applications: Challenges and future research directions”, v zborniku: Automation of Software Test (AST), 2012 7th International Workshop, str. 29–35, 2012.
 - [9] T. Gorschek, E. Tempero, L. Angelis, “On the use of software design models in software development practice: An empirical investigation”, v zborniku: The Journal of Systems & Software, št. 95, str 176–193, 2014.
 - [10] M. Benaroch, Y. Lichtenstein, K. Robinson, “Real options in information technology risk management: An empirical validation of risk-option relationships”, v zborniku: MIS Quarterly: Management Information Systems, št. 30, zv. 4, str. 827–865, 2006.
 - [11] K.-K. Hong, Y.G. Kim, “The critical success factors for ERP implementation: An organizational fit perspective”, v zborniku: Information and Management, št. 40, zv. 1, str. 25–40, 2002.
 - [12] S. L. Vrhovec, T. Hovelja, D. Vavpotič, M. Krisper, “Diagnosing organizational risks in software projects: Stakeholder resistance”, v zborniku: International Journal of Project Management, št. 33, zv. 6, str. 1262–1273, 2015.
 - [13] K. Jehn, E. Mannix, “The dynamic nature of conflict: A longitudinal study of intragroup conflict and group performance”, v zborniku: Academy of Management Journal, št. 44, zv. 2, str. 238–251, 2001.
 - [14] D. Harter, M. Krishnan, S. Slaughter, “Effects of Process Maturity on Quality, Cycle Time, and Effort in Software Product Development”, v zborniku: Management Science, št. 46, zv. 4, str. 451–466, 2000.
 - [15] G. Dietrich, D. Walz, J. Wynekoop, “The failure of SDT diffusion: A case for mass customization”, v zborniku: IEEE Transactions on Engineering Management, št. 44, zv. 4, str. 390–398, 1997.

-
- [16] E. Rogers, “Diffusion of Innovations”, Free Press, New York, 2003.
- [17] C. K. Riemenschneider, B. C. Hardgrave, F. D. Davis, “Explaining software developer acceptance of methodologies: A comparison of five theoretical models”, v zborniku: IEEE Transactions of software engineering, št. 28, zv. 12, str. 1135–1145, 2002.
- [18] F. Davis, “Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology”, v zborniku: MIS Quarterly, št. 13, zv. 3, str. 319–340, 1989.
- [19] F. Davis, R. Bagozzi, P. Warshaw, “User Acceptance of Computer Technology: A Comparison of Two Theoretical Models”, v zborniku: Management Science, št. 35, zv. 8, str. 982–1003, 1989.
- [20] M. Fishbein, I. Ajzen, “Belief, Attitude, Intention and Behaviour: An Introduction to Theory and Research”, Addison-Wesley, Reading, MA, 1975.
- [21] V. Venkatesh, F. Davis, “A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies”, v zborniku: Management Science, št. 46, zv. 2, str. 186–204, 2000.
- [22] G. Moore, I. Benbasat, “Development of an Instrument to Measure the Perceptions of Adopting an Information Technology Innovation”, v zborniku: Information Systems Research, št. 2, zv. 3, str. 192–222, 1991.
- [23] I. Ajzen, “The theory of planned behaviour, v zborniku: Organizational Behavior and Human Decision Processes”, št. 50, zv. 2, str. 179–211, 1991.
- [24] R. Thompson, C. Higgins, J. Howell, “Personal Computing: Toward a Conceptual Model of Utilization”, v zborniku: MIS Quarterly, št. 15, zv. 1, str. 125–143, 1991.

-
- [25] D. Vavpotič, M. Bajec, “An approach for concurrent evaluation of technical and social aspects of software development methodologies”, v zborniku: Information and Software Technology, št. 51, zv. 2, str. 528–545, 2009.
- [26] P. Kruchten, “The Rational Unified Process”, Addison-Wesley, Reading, MA, 1999.
- [27] M. Paulk, C. Weber, S. Garcia, M. B. Chrissis, M. Bush, “Key Practices of the Capability Maturity Model, Version 1.1”, tehnično poročilo, CMU/SEI-93-TR-25, Software Engineering Institute, Carnegie Mellon University, 1993.
- [28] CMMI Product Team, “CMMI for Development, Version 1.3”, tehnično poročilo, CMU/SEI-2010-TR-033, Software Engineering Institute, Carnegie Mellon University, 2010.
- [29] TMMi foundation, “Test Maturity Model Integration (TMMi), release 1.0”, 2012.
- [30] I. Burnstein, “Practical Software Testing: A Process-oriented Approach”, Springer Inc., New York, 2003.
- [31] A. Farooq, “An evaluation framework for software test processes”, doktorska disertacija, Otto von Guericke University, 2009.
- [32] Guru 99, “Static Vs Dynamic Testing”. Dostopno na: <http://www.guru99.com/static-dynamic-testing.html>.
- [33] IEEE computer Society, “IEEE 610-1990 – IEEE Standard Computer Dictionary: A Complitation of IEEE Standard Computer Glossaries”, 1990.
- [34] ISTQB Exam Certification, “What is fundamental test process in software testing?”. Dostopno na: <http://istqbexamcertification.com/what-is-fundamental-test-process-in-software-testing/>.

-
- [35] International Software Testing Qualifications Board, “ISTQB Glossary”. Dostopno na: <https://www.astqb.org/glossary/search/test%20plan>
 - [36] Software Testing Class, “What is difference between Test Cases vs Test Scenarios?”. Dostopno na <http://www.softwaretestingclass.com/what-is-difference-between-test-cases-vs-test-scenarios/>.
 - [37] IEEE computer Society, “IEEE 829-1998 – IEEE Standard for Software and System Test Documentation”, 1998.
 - [38] K. Naik and P. Tripathy, “Software Testing and Quality Assurance: Theory and Practice”, Wiley & Sons, Inc., 2008.
 - [39] Software Testing Fundamentals, “Unit Testing Fundamentals”. Dostopno na: <http://softwaretestingfundamentals.com/unit-testing/>.
 - [40] R. Craig, S. Jaskiel, “Systematic Software Testing”, Artech House, 2002.
 - [41] Software Testing Fundamentals, “Regression Testing Fundamentals”. Dostopno na: <http://softwaretestingfundamentals.com/regression-testing/>.
 - [42] International Software Testing Qualifications Board, “ISTQB Glossary”. Dostopno na: <http://www.astqb.org/glossary/search/requirements-based%20testing>.
 - [43] International Software Testing Qualifications Board, “ISTQB Glossary”. Dostopno na: <http://www.astqb.org/glossary/search/business-process-based%20testing>
 - [44] J. Nielsen, “Usability Engineering”, Morgan Kaufmann Publishers Inc, 1993.
 - [45] ISO 9241-11:1998, “Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability”, 1998.

-
- [46] D. Amalfitano, A. Fasolino, P. Tramontana, B. Robbins, "Testing Android Mobile Applications: Challenges, Strategies, and Approaches", v zborniku: *Advances in Computers*, št. 89, str. 1–52.
- [47] R. Ishida, S. Miller, "Localization vs Internationalization", 2005. Dostopno na: <https://www.w3.org/International/questions/qa-i18n>.
- [48] OpenSignal, "Android Fragmentation Visualized (August 2015)", poročilo. Dostopno na: <http://opensignal.com/reports/2015/08/android-fragmentation/>.
- [49] P. Morville, "User Experience Design", 2004. Dostopno na: http://semanticstudios.com/user_experience_design/.
- [50] ISO 9241-210:2010, "Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems", 2010.
- [51] Kony, "Bridging the Gap: Mobile App Design & Development", Bela knjiga, 2014.
- [52] ISTQB Exam Certification, "What are the different types of software testing tools?". Dostopno na <http://istqbexamcertification.com/what-are-the-different-types-of-software-testing-tools/>.
- [53] T. Koomen, M. Pol, "Test Process Improvement. A practical step-by-step guide to structured testing", Addison-Wesley, 1999.
- [54] M. Tiitinen, "Key Factors for selecting software testing tools", magistrsko delo, Helsinki Metropolia University of Applied Sciences, 2013.
- [55] ISTQB Exam Certification, "What is Incident management tools?". Dostopno na: <http://istqbexamcertification.com/what-is-incident-management-tools/>.
- [56] ISTQB Exam Certification, "What is Review process support tools?". Dostopno na: <http://istqbexamcertification.com/what-is-review-process-support-tools/>.

-
- [57] ISTQB Exam Certification, “What are static analysis tools?”. Dostopno na: <http://istqbexamcertification.com/what-are-static-analysis-tools/>.
- [58] ISTQB Exam Certification, “What is Modelling tools in software testing?”. Dostopno na: <http://istqbexamcertification.com/what-is-modelling-tools-in-software-testing/>.
- [59] ISTQB Exam Certification, “What is Test design tools in software testing?”. Dostopno na: <http://istqbexamcertification.com/what-is-test-design-tools-in-software-testing/>.
- [60] ISTQB Exam Certification, “What is Test data preparation tools in software testing?”. Dostopno na: <http://istqbexamcertification.com/what-is-test-data-preparation-tools-in-software-testing/>.
- [61] ISTQB Exam Certification, “What is Test execution tools in software testing?”. Dostopno na: <http://istqbexamcertification.com/what-is-test-execution-tools-in-software-testing/>.
- [62] ISTQB Exam Certification, “What is Test harness/ Unit test framework tools in software testing?”. Dostopno na: <http://istqbexamcertification.com/what-is-test-harness-unit-test-framework-tools-in-software-testing/>.
- [63] ISTQB Exam Certification, “What is Test comparators in software testing?”. Dostopno na: <http://istqbexamcertification.com/what-is-test-comparators-in-software-testing/>.
- [64] ISTQB Exam Certification, “What is Coverage measurement tools in software testing?”. Dostopno na: <http://istqbexamcertification.com/what-is-coverage-measurement-tools-in-software-testing/>.

-
- [65] ISTQB Exam Certification, “What are Security testing tools in software testing?”. Dostopno na: <http://istqbexamcertification.com/what-is-security-tools-in-software-testing/>.
- [66] ISTQB Exam Certification, “What is Dynamic analysis tools in software testing?”. Dostopno na: <http://istqbexamcertification.com/what-is-dynamic-analysis-tools-in-software-testing/>.
- [67] ISTQB Exam Certification, “What is Performance testing, Load testing and stress-testing tools in software testing?”. Dostopno na: <http://istqbexamcertification.com/what-is-performance-testing-load-testing-and-stress-testing-tools-in-software-testing/>.
- [68] ISTQB Exam Certification, “What is Monitoring tools in software testing?”. Dostopno na: <http://istqbexamcertification.com/what-is-monitoring-tools-in-software-testing/>.
- [69] J. McIver, E. Carmines, “Undimenstional Scaling”, Sage Publications, 1981.
- [70] R. Atkinson, “Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria”, v zborniku: International Journal of Project Management, št. 17, zv. 6, str. 337–342, 1999.
- [71] W. DeLone, E. McLean, “The DeLone and McLean Model of Information Systems Success: A Ten-Year Update”, v zborniku: Journal of Management Information Systems, št. 19, zv. 4, str. 9–30, 2003.
- [72] S. Petter, W. DeLone, E. McLean, “Measuring information system success: models, dimensions, measures and interrelationship”, v zborniku: European Journal of Information Systems, št. 17, zv. 3, str. 236–263, 2008.
- [73] K. Boulding, “Conflict and defense, a general theory”, Harper & Row, 1963.

-
- [74] Wikipedia, “Likert scale”. Dostopno na: https://en.wikipedia.org/wiki/Likert_scale.
- [75] R. Yin, “Case Study Research: Design and Methods”, 4th Edition, SAGE Publications, 2009.
- [76] lka. Dostopno na: <https://www.lka.si/>.
- [77] IBM SPSS Statistics. Dostopno na: <https://www.ibm.com/marketplace/cloud/statistical-analysis-and-reporting/us/en-us>.
- [78] “Innovation Cloud”. Dostopno na: <http://innovationcloud.com/>.
- [79] “InVision”. Dostopno na: <https://www.invisionapp.com/>.
- [80] “Appium”. Dostopno na: <http://appium.io/>.
- [81] “TestRail”. Dostopno na: <http://www.gurock.com/testrail/>.
- [82] “Genymotion”. Dostopno na: <https://www.genymotion.com/>.